**Undergraduate Research Opportunity
Programme in Science**

Financial Mathematics With Python

WANG ZEXIN

ZHOU CHAO

Department of Mathematics

National University of Singapore

AY2016/17 Semester 2

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. In the possiblity of an extension, our project would focus on the incorporation of assumptions beyond the Black-Scholes world into the existing valuation schemes in the current analytics library, and to explore the development of valuation packages for volatility options.

Introduction

The 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.

Regarding the approximation of derivatives prices, both the numerical PDE methods and the Monte Carlo simulations are commonly used in practice. For the pricing of newly structured products, we would prefer to concurrently use the two schemes in order to check for the accuracy of the approximated prices. Valuation of the the options which has readily available PDE discretization methods will be mostly based on the results from the numerical PDE methods. There are several different numerical methods to solve the partial differential equations. Finite difference methods are the oldest and most frequently used, and will be one of our focus in this paper.

The Monte Carlo simulations have been in the favour of many institutions in the finance industry for its compatiability with most of the financial derivatives. Recent development of the parallel computing and GPU has enabled users to generate more data points for the significance of reduction in computational costs of Monte Carlo simulations. Designated variance reduction techniques have been put in place for the simulations, which is also of our interest in the later sections.

Apart from the valuation schemes mentioned above, the derivatives analytics library has already developed certain adaption for the models with assumptions beyond the Black-Scholes world. Discussions on the incorporation of these assumptions into the existing valuation schemes are not included in this paper. Without doubt, efforts in this field to make the analytics library more generic towards the open for extension and closed for modification stage can be rewarding.

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 It$\hat{o}'$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 It$\hat{o}'$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 using 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.

In this paper, we have only implemented one-factor models and two-factor models, in which the original Black-Scholes model, Merton's Jump-diffusion model, and Heston's Stochastic Volatility model can be incorporated. With these models above, we are able to give valuation schemes for European options, Barrier options, Asian options. However the more complicated options such as Multi-asset options will require the usage of multiple-factor models. If we are to plot out the option prices in a diagram, there should be a 3D diagram with y-axis being the underlying prices, and x-axis being the time, omitting the scenario when the option price is dependent on other stochastic terms such as volatility. 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]

One factor model using Explicit Euler Scheme

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

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

One factor model using Implicit Euler Scheme

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

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

Crank-Nicolson method

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. However, people have already explored its strenght upon the valuation using high-dimensional scheme. When doing option pricing with Monte Carlo simulations, we simulate a series of possible scenarios and use the payoff function to calculate one option payoff. By repeating this process N times and take the average of all the discounted option payoffs, a more accurate option price can be obtained. When there is time variation in parameters of the model, or the option payoff is path dependent, or there are a number of stochastic variables that the option price depends on, a finite difference scheme will be both space and time consuming. With the advancements of GPU and parallel computing, just as suggestd in the book "*Python for Finance*",[1] the power of Monte Carlo is able to emerge.

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.

Here we put down the algorithm for Monte Carlo simulations of option prices using underlying prices as control variates. Denote the underlying prices by U_i, payoff function by f, option payoffs by P_i and the option premium by $\bar{P}(\lambda)$.

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 $\mathrm{P}(Y \in \cup_i A_i) = 1$.
By Bayes$'$ Theorem, $\mathrm{E}[Y] = \sum_{i=1}^{n} \mathrm{P}(Y \in A_i)\mathrm{E}[Y|Y \in A_i] = \sum_{i=1}^{n} p_i \mathrm{E}[Y|Y \in A_i]$
We shall exercise proportional sampling which is the simplest case, ensuring $p_i = \mathrm{P}(Y \in A_i)$, the fraction of observations drawn from stratum A_i exactly matches theoretical probability.
Unbiased estimator of $\mathrm{E}(Y)$ is given by $\hat{Y} = \sum_{i=1}^{K}(p_i \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij}) = \frac{1}{n} \sum_{i=1}^{K} \sum_{j=1}^{n_i} Y_{ij}$

$$\mathrm{E}[\hat{Y}] = \mathrm{E}[\sum_{i=1}^{K}(p_i \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij})] = \sum_{i=1}^{K} \mathrm{P}(Y \in A_i)\mathrm{E}[Y|Y \in A_i] = \mathrm{E}[Y]$$

We shall generate a stratification variable X which take values in the union of the disjoint sets A_is. As supposedly the values of X are high correlated to the values of Y, in many cases, Y is a function of X. Let X be the discrete path of underlying asset prices, with Y being the European call payoff discounted to time 0. If X has h discrete prices on one path, $\Omega \in \mathbb{R}^h$ is the sample space for the X_i's, where $\Omega = \cup_i A_i$ is the union of disjoint sets.
For the purpose of stratified sampling, $\mathbb{P}(X \in A_i)$ and $Y|X \in A_i$ should be easy to generate.
In order to further simplify the trouble of dividing Ω into disjoint subsets, we are going to use the stock prices at time $\frac{T}{2}$ as the condition to determine the distributions of A_i. The stock prices at time $\frac{T}{2}$ will be computed based on the Geometric Brownian Motion of the underlying stock prices. Given that we are to produce n equally probable subsets of Ω, the standard normal distribution can be sliced into n pieces to provide sources of generation. With the underlying asset prices at time $\frac{T}{2}$, we can generate k prices at time T and calculate the respective option payoffs. The average payoffs for each stratum A_i is then computed and discounted back to time $\frac{T}{2}$, at which we calculate the average payoffs across the stratum and discount to time 0.

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 changing measures.
We are going to estimate $\alpha = \mathrm{E}[h(X)] = \displaystyle\int h(x)f(x)dx.$

From Monte Carlo simulations, $\hat{\alpha} = \hat{\alpha}(n) = \frac{1}{n}\sum_{i=1}^{n} h(X_i)$

By using a change of measure based on the following assumption:

$$\forall x \in \mathbb{R}^d, f(x) > 0 \implies g(x) > 0$$

Now the estimate becomes $\alpha = \mathrm{E}[h(X)] = \mathrm{E}[h(X)\frac{f(X)}{g(X)}] = \int h(x)\frac{f(x)}{g(x)}dx$

As for simulation, $\hat{\alpha}_G = \frac{1}{n}\sum_{i=1}^{n} h(X_i)\frac{f(X_i)}{g(X_i)}$ We are choosing g to make $X \in A$ more likely after changing measure.

In order to reduce variance, we need to implicitly find the g which gives the minimum possible variance. According to the variance formula $\mathrm{Var}(X) = \mathrm{E}(X^2) - (\mathrm{E}(X))^2$, we have

$$\begin{aligned}\mathrm{Var}^G[h(X)\frac{f(X)}{g(X)}] &= \mathrm{E}^G[h(X)^2\frac{f(X)^2}{g(X)^2}] - (\mathrm{E}^G[h(X)\frac{f(X)}{g(X)}])^2 \\ &= \int \frac{h(x)^2 f(x)^2}{g(x)}dx - (\int \frac{h(x)f(x)}{g(x)}g(x)dx)^2\end{aligned}$$

If $h(x) = \frac{g(x)f(x)}{\mathrm{E}(g(x))}$, the variance of estimate will become 0.

Consider the scenario in which we simulate discrete price paths $S(t_i), \forall i = 0, 1, \ldots, m$, which is assumed to be a Markov Chain with homogeneous property. Let the continuous transition probability be $f_i(S(t_{i-1}), S(t_i))$.

We shall use likelihood ratio $\prod_{i=1}^{m} \frac{f_i(S(t_{i-1}), S(t_i))}{g_i(S(t_{i-1}), S(t_i))}$ as risk-neutral measure.

$$\begin{aligned}\mathbb{E}^{\mathbb{Q}}(X) &= \int xg(x)dx \\ &= \mathbb{E}^{\mathbb{P}}(\frac{d\mathbb{Q}}{d\mathbb{P}}X) \\ &= \mathbb{E}^{\mathbb{P}}(e^{\lambda W_T - \frac{1}{2}\lambda^2 T}X)\end{aligned}$$

In the Black-Scholes world,

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

Hence $\frac{d\mathbb{Q}}{d\mathbb{P}} = exp\{AW_T - \frac{1}{2}A^2 T\}$

European call options

Closed-form formula

$$c_t = S_t N(d_1) - Ke^{-r\tau}N(d_2)$$

where,

$$d_2 = \frac{\ln \frac{S_t}{K} + (\mu - \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}$$

$$d_1 = d_2 + \sigma\sqrt{\tau}$$

Also, N is the cumulative density function of standard normal function.

Numerical PDE

When we are valuating the European call option using the above stated one factor explicit Euler scheme or Crank-nicolson scheme, the 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.

A tabulation of the numercial PDE valuation results for the European call option with current underlying asset price at 100, time 1 to maturity, in a market environment with interest rate 0.05 and volatility 0.25 is shown below: From the tabulated results, we can observe that the Crank-Nicolson method has superiority in accuracy for pricing European call options with relatively lower strike prices, for the ones with high strike prices compared to the current underlying prices, both valuation schemes may not generate a decent result.

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.

Besides the tabulation, there should be other means for us to check upon the convergence of this valuation scheme in the practice of pricing European call options, and whether the variance reduction techniques really work well in improving accuracy of the estimates. Henceforth we fix the strike price at 110, current underlying asset price at 100, at the market condition with interest rate at 0.05, and volatility at 0.25. We gradually increase the number of simulations from 100000 to 1000000 in order to better display the effect of variance reduction technique, in particular, control variate.

It is clear that both estimations converge to the right price at 8.02638469 given by the closed-form formula, but with the help of control variate, the simulation result can converge to the correct price at a much higher rate.

European put options

Closed-form formula

$$p_t = Ke^{-r\tau}N(-d_2) - S_tN(-d_1)$$

where,

$$d_2 = \frac{\ln \frac{S_t}{K} + (\mu - \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}$$

$$d_1 = d_2 + \sigma\sqrt{\tau}$$

Also, N is the cumulative density function of standard normal function.

Numerical PDE

When we are valuating the European put option using the above stated one factor explicit Euler scheme or Crank-nicolson scheme, the payoff function $(K - S_T)^+$ 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.

A tabulation of the numercial PDE valuation results for the European put option with current underlying asset price at 100, time 1 to maturity, in a market environment with interest rate 0.05 and volatility 0.25 is shown below: Similar to the previous section on European call options, we can observe that 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, 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.

Similar to the previous section, we wish to check upon the convergence of this valuation scheme in the practice of pricing European put options, and whether the variance reduction techniques really work well in improving accuracy of the estimates. Henceforth we fix the strike price at 110, current underlying asset price at 100, at the market condition with interest rate at 0.05, and volatility at 0.25. We gradually increase the number of simulations from 100000 to 1000000 in order to better display the effect of variance reduction technique, in particular, control variate.

It is clear that both estimations converge to the right price at 12.661621 given by the closed-form formula, but with the help of control variate, the simulation result can converge to the correct price at a much higher rate. Oscillations of the simulated prices are also significantly reduced by control variate.

Barrier option

The continuous barrier option is one of the most frequently traded path-dependent options. This option comes with a same payoff function as the normal European call/put at maturity subject to the additional condition for a prescribed level to be crossed or not by the underlying asset price during the life of option. The barrier options can be categorized into four different types with the name down-and-out, down-and-in, up-and-out, up-and-in.

Joint distribution of Minimum and Terminal Value

We define the minimum value $m_T = \min\limits_{0 \le t \le T} \ln \dfrac{S_t}{S_0}$ and the terminal value $x_T = \ln \frac{S_T}{S_0}$.

Corollary 1.

$$\mathbb{P}(X_T \geq x, m_T \geq m) = N(\frac{\mu T - x}{\sigma \sqrt{T}}) - e^{\frac{2\mu m}{\sigma^2}} N(\frac{2m - x + \mu T}{\sigma \sqrt{T}})$$

Corollary 2.

$$\mathbb{P}(m_T \geq m) = N(\frac{\mu T - m}{\sigma \sqrt{T}}) - e^{\frac{2\mu m}{\sigma^2}} N(\frac{m + \mu T}{\sigma \sqrt{T}})$$

Details of the proofs for the two corollaries are omitted here.

Closed-form formula

The payoff function of a down-and-out call option can express as $(S_T - K)\mathbb{1}_{\mathbb{F}}$, where $\mathbb{1}_{\mathbb{F}}$ is the indicator variable for the

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

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

Monte Carlo Simulation

A full valuation of the continuous Barrier option will require the usage of the closed-form formula. In the case of pricing a down-and-out call option, discrete Monte Carlo valuation scheme will always give a price higher than the right price as there will always be scenarios missed in which the barrier to be breached. Still, even if we can find an increment in the time zone which can make the scenarios missed to be negligible, the computation cost will increase rapidly and we will not be able to generate enough data points for the accuracy of the estimated option price. Upon this predicament, the research results of Professor Steven Kou[4] shed light on this problem as to give a multiplier to the barrier to enable the discrete valuation of the continuous Barrier option. The new barrier according to the research results is given by $B^* = Be^{\beta\sigma\sqrt{\Delta t}}$, where β is a constant with value 0.5826. As the results have been proven both in theory and in practice, we are convinced that the correct pricing can be obtained.

A tabulation of the resultant down-and-out call option prices is shown below, with the leftmost column indicating the strike prices and the uppermost row indicating the barriers, currently stock price at 100, time 1 to maturity, interest rate at 0.05 and volatility of 0.25. We can see from the simulation results that this valuation scheme should be consistent, and the accuracy of the estimate is unaffected by the change in parameters, as the errors of the simulated option price from the right price do not have a constant pattern in distribution.

In order to look further into the convergence of estimates from Monte Carlo simulations, we plot a graph showing how the price of a down-and-out call option obtained from the simultions is associated with the number of simulations done. The strike price and the barrier have been fixed at 110 and 95, while the current underlying price is at 100. For the market environment, we assume interest rate to be at 0.05 and the volatility to be at 0.25. As the computaion costs for conducting control variate is substantial for Barrier options, we do not show the effect of variance reduction in this case.

It is observable that as the number of simulations increases from 5000 to 400000, the difference between the simulated price and right price converges to 0. In light of this argument, we have demonstrated the convergence of this valuation scheme in the practice of pricing Barrier options. This gives us confidence that with the advancement of GPU and parallel computing, we should be able to generate more accurate results for other path-dependent options using this scheme.

Numerical PDE

When we are valuating the down-and-out call option using the Crank-nicolson scheme, the payoff function $(S_T - K)^+$ is applied for the calculations of the boundary conditions at time T. We have also dealt with the situation during the valuation that any grid point with underlying asset price lower than the barrier will be automatically set to 0. Surprisingly, the research results of Professor Kou[4] is still applicable here. With the continuity correction, we can obtain a set of approximate option prices.

A tabulation of the numercial PDE valuation results for the down-and-out call option with current underlying asset price at 100, time 1 to maturity, in a market environment with interest rate 0.05 and volatility 0.25 is shown below: Similar to the previous section on the European options, we can observe that the Crank-Nicolson method is able to generate a considerably accurate resultant price as compared to the correct price given by closed-form formula. Also, the computation costs of Barrier options are generally much higher than those of the European options even in the numerical PDE valuation scheme, as we have to make corrections upon part of the price vector each time after matrix multiplications.

Conclusion

We have categorized the option pricing problems into the closed-form formula, numerical PDE methods and Monte Carlo simulations. For most of the comparatively more exotic options such as American exercise-style options or Asian options with arithmetic mean, it is considered impossible to give a closed-form formula and hence we have no choice but to resort to Monte Carlo simulations or numerical PDE pricing schemes.

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. Still, there are cases in which the numerical PDE valuation schemes can be very difficult to derive and even if we do, it is possible that the discretization errors accumulates significantly along the grid when we conduct the pricing. Also, the computations for numerical PDE schemes become costly in the more high dimensional problems.

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 favour since the time for simulations can be largely shortened.

Besides these valuation methods stated above, those not mentioned in this paper, for example binomial tree methods, still deserve some attention, as we have no idea now whether they will become the core to develop a universally adopted valuation scheme for financial derivatives pricing in general.

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.