A hybrid computational approach for option pricing

Song-Ping Zhu
University of Wollongong, spz@uow.edu.au

Xinjiang He
University of Wollongong, xinjiang@uow.edu.au

A hybrid computational approach for option pricing

Abstract

In this paper, we propose a novel numerical approach for option pricing with the combination of the MC (Monte Carlo) simulation and the PDE (partial differential equation) approach. Our motivation originates from the fact that within a finite life time of an option contract, the underlying price as well as the range of volatility are expected to vary within a relatively small region centered around the current value of the underlying and the volatility and hence there is no need to compute option prices for the underlying and the volatility values beyond this region. Thus, our hybrid approach takes the advantage of both the MC simulation and PDE approach to form an approach that takes the MC simulation as a special case with the region being extremely small and the PDE approach as another special case with the region being extremely large. Through numerical experiments, we demonstrate that such a hybrid approach enhances computational efficiency, while maintaining the same level of accuracy when either the MC simulation or the PDE approach is used alone for the option prices computed within a suitably chosen interested region.

Disciplines

Engineering | Science and Technology Studies

Publication Details

Zhu , S. & He, X. (2018). A hybrid computational approach for option pricing. International Journal of Financial Engineering, 5 (3), 1850021-1-1850021-16.

A hybrid computational approach for option pricing

Song-Ping ZHU [*] Xin-Jiang HE [†]

Abstract

In this paper, we propose a novel numerical approach for option pricing with the combination of the MC (Monte Carlo) simulation and the PDE (partial differential equation) approach. Our motivation originates from the fact that within a finite life time of an option contract, the underlying price as well as the range of volatility are expected to vary within a relatively small region centered around the current value of the underlying and the volatility and hence there is no need to compute option prices for the underlying and the volatility values beyond this region. Thus, our hybrid approach takes the advantage of both the MC simulation and PDE approach to form an approach that takes the MC simulation as a special case with the region being extremely small and the PDE approach as another special case with the region being extremely large. Through numerical experiments, we demonstrate that such a hybrid approach enhances computational efficiency, while maintaining the same level of accuracy when either the MC simulation or the PDE approach is used alone for the option prices computed within a suitably chosen interested region.

AMS(MOS) subject classification.

Keywords. Combination, Monte Carlo, ADI, Efficiency.

[*] School of Mathematics and Applied Statistics, University of Wollongong NSW 2522, Australia.

[†] Corresponding author. School of Mathematics and Applied Statistics, University of Wollongong NSW 2522, Australia.

1 Introduction

Stochastic volatility models have received a lot of attention since modeling volatility with another stochastic process is believed to better capture different properties of underlying returns [11]. However, it needs to be pointed out that the introduction of another stochastic source has also increased the model complexity. As a result, it is very difficult to find closed-form pricing formula for European options, although there are several models, such as the Stein-Stein model [25] and the Heston model [13], for which analytical solutions can be found. For American options and exotic options, the analytical tractability under stochastic volatility models becomes even worse as it is almost impossible to find any analytical solution for the problems associated with these options. Therefore, numerical approaches must be resorted to when pricing options and proposing innovative numerical schemes with high efficiency becomes a key focus of the research in terms of pricing options under stochastic volatility models.

In the literature, a number of different numerical approaches have been proposed and adopted to efficiently price options. One of the most basic ones is the so-called binomial tree approach [8], in which a tree of the underlying price is generated and option prices are evaluated backwards from the expiry time to the current time. A distinguished feature of this particular method is that it is very straightforward to implement. Although this method can clearly reflect the construction of the replicating portfolio, it may cause some biases since the underlying price actually does not discretely change within a two-valued framework. As another basic approach, the MC simulation technique, which directly simulates the dynamics of the underlying price to find option prices without using any boundary conditions, is also quite popular in option pricing [3, 21, 24] as it has two main advantages; the first is that it can easily deal with path dependent options, and it can handle high-dimensional pricing problems in which options are written on multiple underlying assets. However, this approach is quite often “synonymous” to inefficiency in the option pricing world, due to the requirement of a large number of paths being generated to achieve adequate accuracy, especially for low dimensional problems [16]. Being different from the above two methods, both of which directly simulate the underlying price process, the finite difference method, which is based on the discretization of differential operators, can certainly be employed if a PDE (partial different equation) system governing option prices is derived first. Then, depending on how a differential operator is discretized, different schemes, such as the explicit method, implicit scheme, the ADI (alternating direction implicit) method [18], the predictor-corrector scheme [27] and many others, form all the specific approaches under the umbrella of the PDE approach. This method is firstly applied to option pricing by Schwartz [23] and Brennan & Schwartz [4], and it has been widely used by many others [7, 15]. One of the main advantages for the finite difference method is that a set of option prices can be obtained simultaneously for one time step with a relatively small amount of computational time, while an obvious disadvantage is that the imposition of boundary conditions imposed at infinity for any option pricing PDE system demands a truncation of the computational domain to finite one and such a truncation inevitably introduces errors.

It should be further remarked that the entire computational domain still needs to be discretized, even after the replacement of an infinite domain with a finite domain, and such a "global" discretization, like many paths generated in the MC simulations, also wastes computational resources on computing option prices on the grids that one has no interest on. Therefore, on one hand, a large enough computational domain should be preserved so that the truncation errors could minimized, while, on the other hand, the truncated domain needs to be kept reasonably small so that the discretization of the computational domain with a reasonable resolution does not demand huge computational resources.

A natural question is whether or not there is an approach that can avoid the "cons" of both the MC and PDE approaches while keeping all the "pros" of these two approaches. We provide a positive answer to this question by proposing a hybrid approach that is a combination of the MC simulation technique and the PDE approach (which is referred to as "the MCPDE method" hereafter), with an aim of utilizing the strengths of both these approaches for certain types of options. Our motivation originates from the fact that short-term options are dominating in most of derivative markets, which means that the probability for a dramatic change in the underlying price or the volatility taking place within finite life time of an option contract is quite low. Thus, option traders will probably only need to focus on pricing options with the underlying price and the range of volatility being expected to vary within a relatively small region centered around the current value of the underlying and the volatility and hence there is no need to compute option prices for the underlying and the volatility values beyond this region. To take the advantage of this particular need, our newly proposed MCPDE method focused on a reasonably small computation domain (referred to as the "interested domain" hereafter) centered around the current point of the state variable, $M(S_0, v_0)$, as illustrated in Figure 1. Within the interested domain ABCD, the PDE approach is adopted to provide accurate calculations on a set of grid points with decent resolution. On the other hand, the MC simulations are adopted to generate the needed boundary values on the boundary of the interested domain, so that the PDE approach adopted for the computation of option values within the domain can be actually carried out. In this way, computational resources are not wasted on discretizing the domain beyond the interested domain, based on an assumption that the computational resources required to generate the needed boundary values on the boundary of Domain ABCD through the MC simulations are still less than those required by the PDE approach to compute all the grid values outside of Domain ABCD, if the interested domain ABCD is sufficiently small. Such an assumption is reasonable as once can imagine that our hybrid approach takes the MC simulation as a special case, on one hand, when the interested domain shrinks down to a point, while it, on the other hand, takes the PDE approach as another special case when the the interested domain becomes the entire semi-infinite domain in the original problem. Therefore, the efficiency of the MCPDE method must sit somewhere between the MC simulations and the PDE approach, while the values of a small set of option values can be computed within the interested domain. It should be pointed out that our proposed MCPDE method is completely different from the mixed PDE/MC method proposed in [20], where the stochastic volatility is simulated with the MC simulations and then European option prices are determined through the pricing formula for the Black-Scholes model with a time-dependent volatility.

In order to verify this concept and show the accuracy of the newly proposed approach, we use the pricing of European options under the Heston model as a benchmark example since it possesses a closed-form pricing formula for European options. To demonstrate the efficiency of the newly proposed hybrid approach, we take a special PDE approach, the ADI approach, as an example for the PDE implemented inside an interested domain, and compare the computational times needed to obtain option prices with the same resolution of an interested domain with purely ADI method, purely MC simulation and the MCPDE method.

The rest of the paper is organized as follows. In Section 2, a new numerical approach combining Monte-Carlo simulation and the ADI method is introduced. In Section 3, numerical experiments are carried out to show the accuracy as well as efficiency of the new approach, followed by some concluding remarks given in the last section.

2 A new numerical approach

In this section, we formally introduce the MCPDE method to evaluate options. It should be particularly emphasized that this approach can be easily extended to other stochastic volatility models, and we choose the Heston model as an example to illustrate the newly proposed approach since it possesses a closed-form pricing formula for European options, with which its accuracy can be easily assessed.

To start, let us denote S_t and v_t as the underlying price and the volatility respectively. Then, the dynamics of the Heston model under the risk-neutral measure can be specified

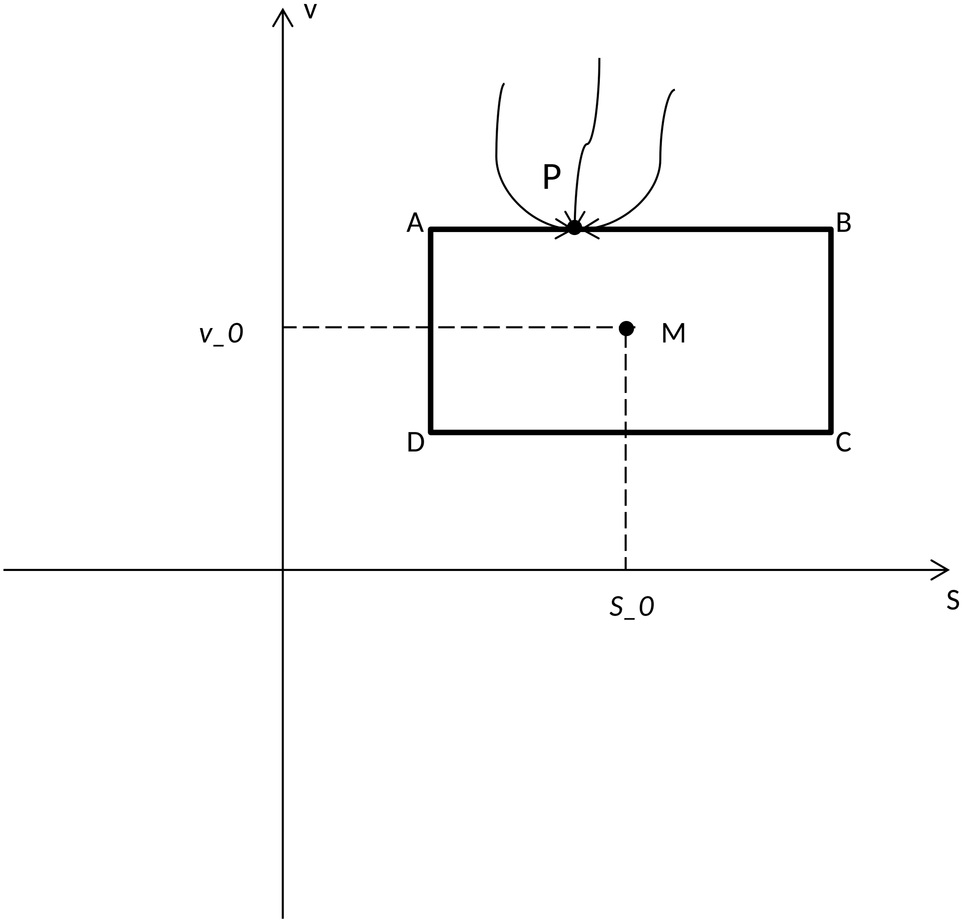

Figure 1: A schematic diagram.

as

$$
\begin{aligned}
\frac{dS_t}{S_t} &= rdt + \sqrt{v_t}dW_{1,t}, \\
dv_t &= k(\beta - v_t)dt + \sigma\sqrt{v_t}dW_{2,t},
\end{aligned}
\tag{2.1}
$$

where r is the risk-free interest rate, k is the mean-reverting speed, β is the mean-reverting level, and σ is the volatility of volatility. W_1 and W_2 are two standard Brownian motions with correlation ρ. If we further let $U(S, v, t)$ denote the price of a European call option, then the governing PDE for U has been shown in many previous papers (e.g., see [12, 13])

$$
\frac{\partial U}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 U}{\partial S^2} + \rho\sigma vS\frac{\partial^2 U}{\partial S\partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 U}{\partial v^2} + rS\frac{\partial U}{\partial S} + k(\beta - v)\frac{\partial U}{\partial v} - rU = 0. \tag{2.2}
$$

Introducing a new variable, the time to expiry $\tau = T - t$, the terminal value problem can be transformed into an initial value problem. On the other hand, to eliminate the variable coefficients in PDE (2.2), we adopt the log-transform as $x = \log(S)$. Therefore, PDE (2.2) can be transformed into

$$\frac{\partial U}{\partial \tau} = AU, \tag{2.3}$$

where the operator is defined as

$$A = a(v)\frac{\partial^2}{\partial x^2} + b(v)\frac{\partial^2}{\partial v^2} + c(v)\frac{\partial^2}{\partial x \partial v} + d(v)\frac{\partial}{\partial x} + e(v)\frac{\partial}{\partial v} - r, \tag{2.4}$$

with

$$a(v) = \frac{1}{2}v,\ b(v) = \frac{1}{2}\sigma^2 v,\ c(v) = \rho\sigma v,\ d(v) = r - \frac{1}{2}v,\ e(v) = k(\beta - v).$$

Here, the initial condition is $U(x, v, \tau)|_{\tau=0} = \max(e^x - K, 0)$. Usually, the domain $[-\infty, +\infty] \times [0, +\infty]$ should be truncated as $[-X_{max}, X_{max}] \times [0, V_{max}]$ with X_{max} and V_{max} chosen to be large enough so that the boundary conditions at infinity can be regarded as a good approximation for those at the truncated points. However, as stated above, this can waste computational time on computing values of the unknown function in the area that we have no interest in.

In contrast, our MCPDE approach combines the MC simulation method and the PDE method to avoid such kind of waste. In particular, it directly generates the boundary value for our interested domain by simulation with the MC method so that we will be able to save certain amount of time. If we denote the interested domain as $[X_1, X_2] \times [V_1, V_2]$, then the boundary values we need to generate are actually $U(x, V_1, \tau)$, $U(x, V_2, \tau)$, $U(X_1, v, \tau)$ and $U(X_2, v, \tau)$. It should be pointed out that simulating the Heston model with the MC approach is an interesting problem itself due to the presence of the Cox-Ingersoll-Ross process, and a number of different schemes are proposed, such as the coordination changing scheme [19], the exact simulation scheme [5, 6] and so on. We refer interested readers to [22] for a detailed comparison of several different schemes. Euler schemes are also quite popular since it is simple and efficient to implement, and even in the case that a more efficient exact simulation approach would be established for the Heston model in the future, it will still remain useful for strongly path-dependent options and stochastic volatility extension of certain markets [1, 2]. Thus, what we adopt here is one particular kind of Euler schemes proposed by Higham & Mao [14], which is specified as

$$\begin{aligned} S_{t+\Delta t} &= S_t + rS_t dt + \sqrt{|v_t|} S_t dW_{1,t}, \\ v_{t+\Delta t} &= v_t + k(\beta - v_t)\Delta t + \sigma\sqrt{|v_t|} dW_{2,t}, \end{aligned} \tag{2.5}$$

where $|\cdot|$ means taking the absolute value.

With the initial condition and all the boundary values in hands, we are able to price options in the interested domain with PDE approaches. In the following, we choose one kind of the most popular PDE approaches, the ADI method, since it is a very useful tool for solving parabolic equations on rectangular domains. Generally speaking, it is of great efficiency since it can reduce a two-dimensional problem to a succession of many one-dimensional problems, whose final matrix is tridiagonal and can thus be easily solved. Before the ADI method is formally applied, the time derivative of U should be discretized in advance. We first split the operator A as

$$A = A_0 + A_1 + A_2, \tag{2.6}$$

such that A_0, A_1 and A_2 represent the mixed derivative, the spatial derivative in the x direction and the spatial derivative in the v direction respectively, i.e.,

$$\begin{aligned} A_0 &= c(v)\frac{\partial^2}{\partial x \partial v}, \\ A_1 &= a(v)\frac{\partial^2}{\partial x^2} + d(v)\frac{\partial}{\partial x} - \frac{1}{2}r, \\ A_2 &= b(v)\frac{\partial^2}{\partial v^2} + e(v)\frac{\partial}{\partial v} - \frac{1}{2}r. \end{aligned}$$

Then, if a uniform discretization in the τ direction is performed with the step size being $\Delta\tau = \dfrac{T}{N_\tau}$, applying the first-order implicit Euler scheme to PDE (2.3) yields

$$\frac{U^{n+1} - U^n}{\Delta\tau} = (A_0 + A_1 + A_2)U^{n+1} + \mathbb{O}(\Delta\tau),$$

with $U^n = U(x, v, n\Delta\tau)$, which can be rearranged as

$$[I - \Delta\tau(A_0 + A_1 + A_2)]U^{n+1} = U^n + \mathbb{O}(\Delta\tau^2).$$

Similarly, the first-order explicit Euler scheme of (2.3) can be derived as

$$U^{n+1} = [I + \Delta\tau(A_0 + A_1 + A_2)]U^n + \mathbb{O}(\Delta\tau^2).$$

As a result, the weighted average of implicit and explicit scheme can be obtained

$$[I - \theta\Delta\tau(A_0 + A_1 + A_2)]U^{n+1} = [I + (1-\theta)\Delta\tau(A_0 + A_1 + A_2)]U^n + \mathbb{O}(\Delta\tau^2). \qquad (2.7)$$

It should be remarked that if $\theta = 0$ or $\theta = 1$, (2.7) degenerates to the explicit scheme or implicit scheme, respectively. When θ takes the value of $\dfrac{1}{2}$, (2.7) becomes the Crank–Nicolson scheme. In order to solve this problem separately in two directions with the ADI method, (2.7) is further expressed as

$$\begin{aligned}(I - \theta\Delta\tau A_1)(I - \theta\Delta\tau A_2)U^{n+1} &= [I + \Delta\tau A_0 + (1-\theta)\Delta\tau A_1 + (1-\theta)\Delta\tau A_2 + \theta^2(\Delta\tau)^2 A_1 A_2]U^n \\ &+ [\theta\Delta\tau A_0 + \theta^2(\Delta\tau)^2 A_1 A_2](U^{n+1} - U^n) + \mathbb{O}(\Delta\tau^2). \qquad (2.8)\end{aligned}$$

Since the order of $U^{n+1} - U^n$ is $\mathbb{O}(\Delta\tau)$, $[\theta\Delta\tau A_0 + \theta^2(\Delta\tau)^2 A_1 A_2](U^{n+1} - U^n)$ can certainly be merged into the error term and thus we can obtain

$$(I - \theta\Delta\tau A_1)(I - \theta\Delta\tau A_2)U^{n+1} = [I + \Delta\tau A_0 + (1-\theta)\Delta\tau A_1 + (1-\theta)\Delta\tau A_2 + \theta^2(\Delta\tau)^2 A_1 A_2]U^n + \mathbb{O}(\Delta\tau^2).$$

Therefore, we finally arrive at the finite difference equation

$$(I-\theta\Delta\tau A_1)(I-\theta\Delta\tau A_2)U^{n+1} = [I+\Delta\tau A_0+(1-\theta)\Delta\tau A_1+\Delta\tau A_2]U^n-(I-\theta\Delta\tau A_1)\theta\Delta\tau A_2U^n, \tag{2.9}$$

if we omit the terms of order $\mathbb{O}(\Delta\tau^2)$.

With Equation (2.9) in hands, we are now ready to proceed to the ADI method. In fact, there are different schemes, such as the Craig-Sneyd scheme [9] and the Hundsdorfer-Verwer scheme [17]. What we adopt here is the Douglas-Rachford scheme [10] since it is a two-step scheme, which is more convenient to be implemented. In particular, the first step of the Douglas-Rachford scheme for our case is to compute an intermediate variable (we denote it as Y) from

$$(I-\theta\Delta\tau A_1)Y = [I+\Delta\tau A_0+(1-\theta)\Delta\tau A_1+\Delta\tau A_2]U^n, \tag{2.10}$$

by fixing the variable in the v direction. If the interested domain $[X_1, X_2] \times [V_1, V_2]$ is discretized with the number of steps in the x and v direction being set to be N_x and N_v respectively, the value of a European option at a grid point can be expressed as

$$U(x, v, \tau) = U(i\Delta x, j\Delta v, n\Delta\tau) = U^n_{i,j}. \tag{2.11}$$

Moreover, the first-order and the second-order non-cross spacial derivatives are approximated by the standard central difference scheme, while the crossed spatial derivative is calculated as

$$\frac{\partial^2 U^n_{i,j}}{\partial x\partial v} = \frac{\dfrac{U^n_{i+1,j+1}-U^n_{i-1,j+1}}{2\Delta x}-\dfrac{U^n_{i+1,j-1}-U^n_{i-1,j-1}}{2\Delta x}}{2\Delta v}.$$

Therefore, Equation (2.10) can be split into $(N_v - 1)$ algebraic equations, the matrix form of which can be derived as

$$B^1 Y_j = P_j^1 + R_j^1,\ j = 1, 2, ..., N_v - 1, \tag{2.12}$$

where R_j^1 is a $(N_x - 1)$-dimensional row vector, whose first element and last element are $\theta \Delta\tau(\frac{a_j}{\Delta x^2} - \frac{d_j}{2\Delta x})Y_{0,j}$ and $\theta \Delta\tau(\frac{a_j}{\Delta x^2} + \frac{d_j}{2\Delta x})Y_{N_x,j}$, respectively, while other elements take the value of zero. $Y_j = (Y_{1,j}, Y_{2,j}, ..., Y_{N_x-1,j})'$, $P_j^1 = (p_{1,j}^1, p_{2,j}^1, ..., p_{N_x-1,j}^1)'$ with $p_{i,j}^1$ expressed as

$$\begin{aligned} p_{i,j}^1 &= U_{i,j}^n + c_j \Delta\tau \frac{U_{i+1,j+1}^n - U_{i+1,j-1}^n - U_{i-1,j+1}^n + U_{i-1,j-1}^n}{4\Delta x \Delta v} \\ &+ (1-\theta)\Delta\tau(a_j \frac{U_{i+1,j}^n - 2U_{i,j}^n + U_{i-1,j}^n}{\Delta x^2} + d_j \frac{U_{i+1,j}^n - U_{i-1,j}^n}{2\Delta x} - \frac{1}{2} r U_{i,j}^n) \\ &+ \Delta\tau(b_j \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{\Delta v^2} + e_j \frac{U_{i,j+1}^n - U_{i,j-1}^n}{2\Delta x} - \frac{1}{2} r U_{i,j}^n), \end{aligned} \tag{2.13}$$

and B^1 is a tridiagonal matrix defined as

$$B^1 = \begin{bmatrix} 1+\theta\Delta\tau(\frac{2a_j}{\Delta x^2}+\frac{1}{2}r) & -\theta\Delta\tau(\frac{a_j}{\Delta x^2}+\frac{d_j}{2\Delta x}) & & 0 \\ -\theta\Delta\tau(\frac{a_j}{\Delta x^2}-\frac{d_j}{2\Delta x}) & \ddots & \ddots & \\ & \ddots & \ddots & -\theta\Delta\tau(\frac{a_j}{\Delta x^2}+\frac{d_j}{2\Delta x}) \\ 0 & & -\theta\Delta\tau(\frac{a_j}{\Delta x^2}-\frac{d_j}{2\Delta x}) & 1+\theta\Delta\tau(\frac{2a_j}{\Delta x^2}+\frac{1}{2}r) \end{bmatrix}.$$

Once we have obtained the solution of the intermediate variable Y, we can move on to the second step to find the solution to U^{n+1} through

$$(I - \theta\Delta\tau A_2)U^{n+1} = Y - \theta\Delta\tau A_2 U^n, \tag{2.14}$$

by fixing the variable in the x direction since the operator in the left hand side of Equation

(2.14) is with respect to v only. Similarly, we only need to solve (N_x-1) algebraic equations

$$B^2U_i^{n+1} = P_i^2 + R_i^2,\ i = 1, 2, ..., N_x - 1, \tag{2.15}$$

where all the elements of the row vector R_i^2 are zero except that the first and the last element take the value of $\theta\Delta\tau(\frac{b_j}{\Delta v^2} - \frac{e_j}{2\Delta v})U_{i,0}$ and $\theta\Delta\tau(\frac{b_j}{\Delta v^2} + \frac{e_j}{2\Delta v})U_{i,N_v}$ respectively. U_i^{n+1} and P_i^2 are defined as $(U_{i,1}, U_{i,2}, ..., U_{i,N_v-1})'$, $P_j^2 = (p_{i,1}^2, p_{i,2}^2, ..., p_{i,N_v-1}^2)'$ respectively, with $p_{i,j}^2$ expressed as

$$p_{i,j}^2 = Y_{i,j} - \theta\Delta\tau(b_j\frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{\Delta v^2} + e_j\frac{U_{i,j+1}^n - U_{i,j-1}^n}{2\Delta v} - \frac{1}{2}rU_{i,j}^n). \tag{2.16}$$

The Matrix B^2 is also a tridiagonal matrix and it can be specified as

$$B^2 = \begin{bmatrix} 1+\theta\Delta\tau(\frac{2b_j}{\Delta v^2}+\frac{1}{2}r) & -\theta\Delta\tau(\frac{b_j}{\Delta v^2}+\frac{e_j}{2\Delta v}) & & 0 \\ -\theta\Delta\tau(\frac{b_j}{\Delta v^2}-\frac{e_j}{2\Delta v}) & \ddots & \ddots & \\ & \ddots & \ddots & -\theta\Delta\tau(\frac{b_j}{\Delta v^2}+\frac{e_j}{2\Delta v}) \\ 0 & & -\theta\Delta\tau(\frac{b_j}{\Delta v^2}-\frac{e_j}{2\Delta v}) & 1+\theta\Delta\tau(\frac{2b_j}{\Delta v^2}+\frac{1}{2}r) \end{bmatrix}.$$

By now, we have presented our MCPDE method for option pricing. In summary, once a discretization of $[X_1, X_2] \times [V_1, V_2] \times [0, T]$ is chosen, the first step of our newly proposed approach is to generate boundary values, $U_{0,j}^n$, $U_{N_x,j}^n$, $U_{i,0}^n$ and U_{i,N_v}^n for each $n = 0, 1, ..., N_\tau$, $i = 0, 1, ..., N_x$, and $j = 0, 1, ..., N_v$ by directly simulating Dynamic (2.1) with the MC technique. Then the second step is to work out an intermediate variable Y with Equation (2.12) and the obtained U^n. It should be remarked here that we need to find the boundary values for Y, i.e., $Y_{0,j}$ and $Y_{N_x,j}$ before we are able to solve Equation (2.12). In fact, the calculation of Y on the two boundaries can be dealt with similarly by utilizing Equation

(2.14), which can yield

$$\begin{aligned} Y_0 &= (I - \theta\Delta\tau A_2)U_0^{n+1} + \theta\Delta\tau A_2 U_0^n, \\ Y_{N_x} &= (I - \theta\Delta\tau A_2)U_{N_x}^{n+1} + \theta\Delta\tau A_2 U_{N_x}^n, \end{aligned} \tag{2.17}$$

since we have generated all the boundary values for $U^n, n = 0, 1, ..., N_\tau$. Once we have successfully obtained Y, the last step is to derive U^{n+1} with Equation (2.15).

Once a new numerical approach is proposed, its accuracy needs to be verified first. Moreover, it is also of interest to show whether the new method is advantageous than those existing methods, as far as the computational speed is concerned since our initial aim is to find an alternative method, which can save us some time when pricing options if we only focus on an interested domain with a high resolution. These two issues will be illustrated in the next section.

3 Numerical examples and discussions

In this section, the accuracy of the newly proposed numerical approach is demonstrated by making comparison of our results and those obtained through the closed-form pricing formula for European options in the Heston model. Then, the computational time of our approach, purely Monte Carlo approach and purely ADI approach to work out the same set of option prices is presented for the purpose of assessing efficiency. Unless otherwise stated, the values of parameters we use are listed as follows. The risk-free interest rate r is assumed to be 0.03, while the correlation between the underlying price and the volatility ρ is 0.8. The mean-reverting speed k and level β are set to 2 and 0.5, respectively. The volatility of volatility σ and the initial value of the volatility v_0 are chosen to be of the same value 0.3. The time to expiry T is set to 1, and the number of time steps, N_τ, is 100, while the number of paths in the MC simulation, N_{MC}, is set to be 200,000. The lower bound and the upper bound for the underlying price, X_1 and X_2, are chosen to be log(90) and log(110), respectively. It should be pointed out that in the following, nx and nv are used to denote the number of grid points in the x and v direction, respectively.

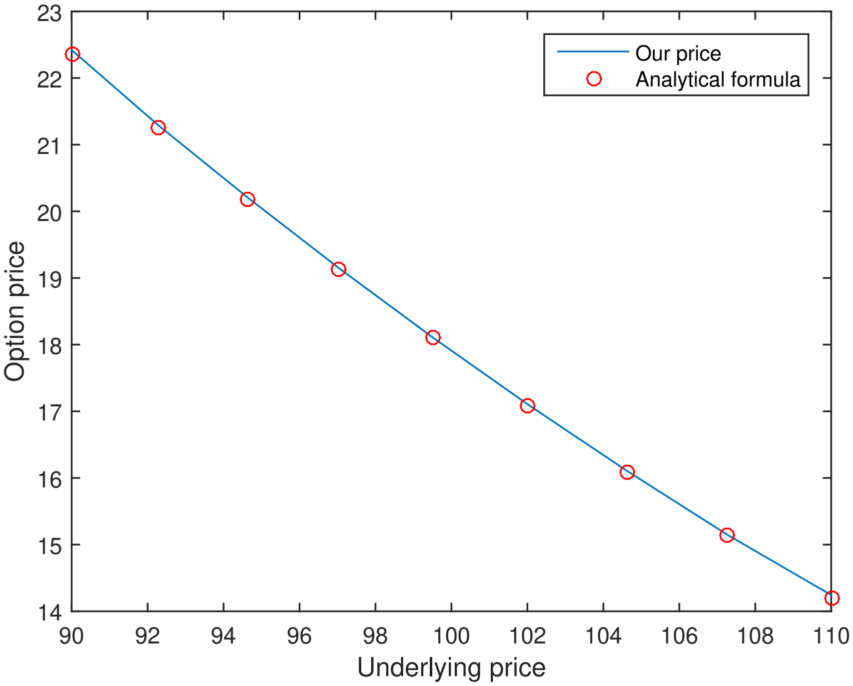

Figure 2: Comparison of European put option prices calculated with our approach and analytical formula

Depicted in Figure 2 are European put option prices calculated from the MCPDE method and the analytical formula. What can be observed first is that option prices are a monotonic decreasing function of the underlying price, which is consistent with the property of European puts. Moreover, we can observe an excellent agreement between option prices calculated with the MCPDE method and the analytical formula; the relative errors between the two prices on the boundaries and internal grid points are approximately 0.3% and less than 0.1%, respectively.

One may be interested in how the accuracy is affected by the number of grids in the x and v directions, and thus we also present the relative errors (based on the difference between the option prices calculated from the MCPDE method and those from the an-

Table 1: Accuracy with different grid resolutions ($S = 99.5$)

	$v_0 = 0.1$	$v_0 = 0.2$	$v_0 = 0.3$	$v_0 = 0.4$	$v_0 = 0.5$
$nv = nx = 5$	14.281(0.03%)	16.337 (0.24%)	18.076 (0.12%)	19.710(0.14%)	21.249 (0.01%)
$nv = nx = 9$	14.296(0.12%)	16.298 (0.01%)	18.125 (0.14%)	19.742(0.03%)	21.258 (0.05%)
$nv = nx = 13$	14.294(0.12%)	16.290 (0.04%)	18.134 (0.19%)	19.734(0.02%)	21.315 (0.32%)
$nv = nx = 17$	14.282(0.03%)	16.289 (0.05%)	18.105 (0.04%)	19.729(0.04%)	21.320 (0.35%)

alytical formula) of numerical results under different grid resolutions in Table 1. In this example, the interested domain is chosen as $[S, v] \in [90, 110] \times [0.1, 0.5]$, which means that the prices in the first and last column of Table 1 are the values on the boundary and were thus generated with Monte Carlo simulation (the number of paths used is fixed to be 200,000). From Table 1, one can indeed conclude that the MCPDE method provides satisfactory results, with the maximum relative error being less than 0.4%, even with the most crude grid resolution when the number of grids in the x and v directions is only 5. One can also observe that while the relative errors on the boundary grids are roughly of the same order, i.e., independent from the grid resolution for the PDE approach being applied inside the interested domain as expected, the relative errors inside the interested domain decrease as the number of grids in the interested domain increases. This suggests that a sufficient grid resolution is needed inside the interested domain, in order to achieve a balanced relative error level between the calculation carried out on the internal grid through the PDE approach and that carried on the boundary grid through the MC simulations. Such a balance is optimal in terms of computational resources not being wasted on either over-accurate boundary values being calculated with “too many” paths in the MC simulations or under-accurate boundary values with unjustified grid resolution inside the interested domain. For this particular example, one can see that by the time the number of spacial interval reaches 16, we have already achieved enough accuracy.

Another interesting topic that is worth of investigation is the influence of the number of paths used in simulating the boundaries on the accuracy of the obtained option prices. In order to demonstrate this point, option prices calculated through the MCPDE method

Table 2: Accuracy with different numbers of paths used in MC simulation ($S = 99.5$)

	$v_0 = 0.1$	$v_0 = 0.2$	$v_0 = 0.3$	$v_0 = 0.4$	$v_0 = 0.5$
$N_{MC} = 100$	12.022(15.80%)	15.651 (3.96%)	17.894 (1.13%)	19.789(0.27%)	19.817 (6.73%)
$N_{MC} = 200$	13.207(7.50%)	16.469 (1.06%)	18.148 (0.27%)	19.481(1.30%)	20.226 (4.80%)
$N_{MC} = 1000$	14.415(0.96%)	16.310 (0.08%)	18.760 (3.66%)	19.9644(1.15%)	20.879 (1.73%)
$N_{MC} = 10000$	14.230(0.33%)	16.263 (0.21%)	18.041 (0.32%)	19.764(0.14%)	21.682 (2.05%)
$N_{MC} = 100000$	14.250(0.19%)	16.302 (0.03%)	18.109 (0.06%)	19.818(0.41%)	21.320 (0.35%)

with different numbers of paths (the interested domain is assumed to be $[S, v] \in [90, 110] \times [0.1, 0.5]$) and the relative errors between the obtained values and those computed from the analytical formula are shown in Table 2. As expected, both relative errors at the boundary and those at the inner grid points generally show a downward trend when the number of paths used in MC simulation of the boundaries is enlarged. In particular, the average relative error for the case of 100 simulation paths is the highest, 5.58%, which decreases to 2.99% and 1.52% when the number of paths increases to 200 and 1,000 respectively. By the time that the number of paths reaches 10,000, the average relative error is already below 1% at only 0.61%, which implies that 10,000 simulation paths would be adequate if one does not require a very high degree of accuracy, and such average relative error further decreases to 0.21% when the number of simulation paths is 100,000.

Table 3: CPU time (seconds) with different approaches

	MCPDE	Monte-Carlo	ADI
Single Point	19	0.8	2
$nv = nx = 5$	19	20.5	2
$nv = nx = 11$	42	99	9
$nv = nx = 51$	204	2127	208
$nv = nx = 101$	407	8471	804
$nv = nx = 201$	805	34721	3337

With the gained confidence of our newly proposed approach in terms of accuracy, we can now proceed to examining computational efficiency. This is achieved through comparing the computational time spent on calculating option prices with a pre-fixed relative error being less than 1%. It should be remarked that although the interested domain is set to be $[S, v] \in [90, 110] \times [0.1, 0.2]$, the operating domain of the purely ADI method is chosen to be $[0, 1800] \times [0, 5]$ to eliminate the effect of the truncation of boundaries [26]. Therefore, in order to have the same grid resolution for the interested domain when using the purely ADI method, the step size in both S and v direction of the operating domain $[0, 1800] \times [0, 5]$ should be the same as that of the interested domain $[90, 110] \times [0.1, 0.2]$. This will result in the grid points in the direction of S and v used in the purely ADI method being respectively 90 and 50 times more than those corresponding ones when using the MCPDE method. In this case, the recorded CPU times with different grid resolution are tabulated in Table 3. Clearly, purely using the MC simulation is advantageous over the other two methods when we are only interested in a small number of points as it only takes 0.8 seconds to obtain one accurate option price compared with 19 seconds of our method and 2 seconds of purely ADI method. The extreme case is when one wants to compute the option with one grid point. In this case, naturally there is no need to use the PDE at all, as the interested domain shrinks to a point. When the option prices need to be simultaneously computed in a finite region in the state variable space, purely MC simulations could be computationally very expensive and its total computational time increases sharply with the interested number of points as shown in Table 3. In fact, more computational time than the other two methods is needed when the grids in the x and v directions of the interested domain exceeds 5, and more than 10 times of the computational time is needed to obtain the same set of option prices if the number of spatial interval is over 50. On the other hand, when the number of grids in the interested domain is still reasonably small, purely ADI method is clearly the most efficient method (take nv and nx being equal to 11 as an example). When the number of spatial interval reaches 50, the computational times associated with the MCPDE method and purely ADI are roughly the same. Then, if one wants to achieve much fine grid resolution within the interested domain for various practical reasons, superiority of the MCPDE method over the other two becomes much clear, and the increasing trend continues as the resolution inside the interested domain is further refined. By the time when the number of spatial interval reaches 200, the computational time associated with the MCPDE method is only a quarter of that associated with the purely ADI method. This has certainly demonstrated competitiveness of our newly proposed approach when option traders need to compute option price in a particular region with high resolution.

4 Conclusion

In this paper, a novel numerical approach that combines both the MC simulation and the PDE approach is proposed. The newly proposed MCPDE approach is suitable for the option pricing task, in which a set of option prices need to be computed within a small region of underlying asset and volatility (or interest rate) values with a reasonable high resolution under a stochastic volatility (or stochastic interest rate) model. Under this method, boundary values of the interested domain are generated with the MC simulation, while option prices inside this region are calculated with the PDE approach. In this way, the discretization with the same resolution as that inside the interested domain is avoided, so that balanced accuracy and efficiency can be achieved for the need of computing option prices on a finite set of grid points. Through numerical experiments, we have demonstrated that the MCPDE approach is indeed superior over purely using Monte Carlo simulation or the ADI approach when the number of spacial interval in the interested domain is over 50.

References

[1] Leif Andersen and Jesper Andreasen. Volatile volatilities. *Risk*, 15:163–168, 2002.

[2] Leif BG Andersen and Rupert Brotherton-Ratcliffe. Extended libor market models with stochastic volatility. *Journal of Computational Finance*, 9:1–40, 2005.

[3] Phelim Boyle, Mark Broadie, and Paul Glasserman. Monte carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8):1267–1321, 1997.

[4] Michael J Brennan and Eduardo S Schwartz. Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial and Quantitative Analysis*, 13(03):461–474, 1978.

[5] Mark Broadie and Ozgur Kaya. Exact simulation of option greeks under stochastic volatility and jump diffusion models. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 2, pages 1607–1615. IEEE, 2004.

[6] Mark Broadie and Özgür Kaya. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations research*, 54(2):217–231, 2006.

[7] Rama Cont and Ekaterina Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential lévy models. *SIAM Journal on Numerical Analysis*, 43(4):1596–1626, 2005.

[8] John C Cox, Stephen A Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979.

[9] Ian JD Craig and Alfred D Sneyd. An alternating-direction implicit scheme for parabolic equations with mixed derivatives. *Computers & Mathematics with Applications*, 16(4):341–350, 1988.

[10] Jim Douglas and Henry H Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2):421–439, 1956.

[11] Patrick S Hagan, Deep Kumar, Andrew S Lesniewski, and Diana E Woodward. Managing smile risk. *The Best of Wilmott*, 1:249–296, 2002.

[12] Xin-Jiang He and Song-Ping Zhu. An alternative form used to calibrate the heston option pricing model. *Computers & Mathematics with Applications*, 71(9):1831–1842, 2016.

[13] Steven L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6(2):327–343, 1993.

[14] Desmond J Higham and Xuerong Mao. Convergence of monte carlo simulations involving the mean-reverting square root process. *Journal of Computational Finance*, 8(3):35–61, 2005.

[15] John Hull and Alan White. Valuing derivative securities using the explicit finite difference method. *Journal of Financial and Quantitative Analysis*, 25(01):87–100, 1990.

[16] John C Hull. *Options, futures, and other derivatives*. Pearson Education India, 2006.

[17] Willem Hundsdorfer and Jan G Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*, volume 33. Springer Science & Business Media, 2013.

[18] KJ In't Hout and S Foulon. Adi finite difference schemes for option pricing in the heston model with correlation. *Int. J. Numer. Anal. Model*, 7(2):303–320, 2010.

[19] Peter Jäckel. *Monte Carlo methods in finance*. J. Wiley, 2002.

[20] Grégoire Loeper and Olivier Pironneau. A mixed pde/monte-carlo method for stochastic volatility models. *Comptes Rendus Mathematique*, 347(9-10):559–563, 2009.

[21] Francis A Longstaff and Eduardo S Schwartz. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14(1):113–147, 2001.

[22] Roger Lord, Remmert Koekkoek, and Dick Van Dijk. A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, 10(2):177–194, 2010.

[23] Eduardo S Schwartz. The valuation of warrants: implementing a new approach. *Journal of Financial Economics*, 4(1):79–93, 1977.

[24] Eduardo S Schwartz and Walter N Torous. Prepayment and the valuation of mortgage-backed securities. *The Journal of Finance*, 44(2):375–392, 1989.

[25] Elias M Stein and Jeremy C Stein. Stock price distributions with stochastic volatility: an analytic approach. *Review of Financial Studies*, 4(4):727–752, 1991.

[26] Paul Wilmott, Jeff Dewynne, and Sam Howison. Option pricing. 1994.

[27] Song-Ping Zhu and Wen-Ting Chen. A predictor–corrector scheme based on the ADI method for pricing American puts with stochastic volatility. *Computers & Mathematics with Applications*, 62(1):1–26, 2011.