J. KSIAM Vol.22, No.1, 75–89, 2018 http://dx.doi.org/10.12941/jksiam.2018.22.075

COMPARATIVE STUDY OF NUMERICAL ALGORITHMS FOR THE ARITHMETIC ASIAN OPTION

JIAN WANG[1] AND JUNGYUP BAN[2], SEONGJIN LEE[2], CHANGWOO YOO[2,†]

[1] DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 02841, KOREA

[2] DEPARTMENT OF FINANCIAL ENGINEERING, KOREA UNIVERSITY, SEOUL 02841, KOREA
E-mail address: coreapoa@korea.ac.kr

ABSTRACT. This paper presents the numerical valuation of the arithmetic Asian option by using the operator-splitting method (OSM). Since there is no closed-form solution for the arithmetic Asian option, finding a good numerical algorithm to value the arithmetic Asian option is important. In this paper, we focus on a two-dimensional PDE. The OSM is famous for dealing with plural-dimensional PDE using finite difference discretization. We provide a detailed numerical algorithm and compare results with MCS method to show the performance of the method.

1. INTRODUCTION

We consider an efficient and accurate finite difference method [1] and Monte Carlo simulation [2] for an arithmetic Asian option. The Asian option is a contract that gives the holder the right to buy an asset based on its average price over some prescribed period of time [3]. There are two types of Asian options such as arithmetic Asian option and geometric Asian option. Geometric Asian option with payoff which depends on geometric mean of underlying asset over time interval has a closed form solution. On the other hand, arithmetic Asian option with payoff which depends on arithmetic mean of underlying asset over time interval does not have a closed form solution. Thus, finding a good numerical algorithm to value arithmetic Asian option is important.

A lot of previous studies for the Asian option have been implemented. There are a number of studies to approximate this option [4, 5, 6, 7]. However, those approximation formula are only suitable for a simple type of Asian option, i.e., European type of Asian option. In general, the value of derivative can be found by solving PDE [8]. There are five representative forms of PDE for Asian option. Two of them have two spatial dimensions and they are derived by Ingersoll

Received by the editors February 28 2018; Accepted March 13 2018; Published online March 16 2018.
2000 *Mathematics Subject Classification.* 93B05.
Key words and phrases. Arithmetic Asian option, operator splitting method, finite difference method, Black–Scholes equation.
† Corresponding author.

in [9] and Duffy in [1]. They are derived by using same hedging argument underlying Black–Scholes in [10]. In this paper, we are going to deal with PDE derived by Ingersoll. Rest of them have one spatial dimension which is derived by change of numeraire technique. Ingersoll set $R = S/I$ and Duffy set $R = I/S$ where S is stock process and $I = \int S dt$. And they work some stochastic calculus to derived deterministic PDE in [1, 9, 11]. Turning upside down of numerator and denominator and some work of stochastic calculus can make PDE more simpler. However, these PDEs are only available for floating strike type of Asian option. In 1995, Rogers and Shi [12] derived one spatial dimensional PDE which is available for both floating strike type and fixed strike type of Asian options by setting $x = [K - \int_0^t S(\tau)\mu(d\tau)]/S_t$. Floating and fixed strikes are as follows: $(A(T) - K)^+$, $(K - A(T))^+$, $(S(T) - A(T))^+$, and $(A(T) - S(T))^+$, respectively, for fixed strike call, fixed strike put, floating strike call, and floating strike put, where $A(T) = I(T)/T$.

Reduction of spatial dimension for Asian option makes PDE more simpler. Also, when we solve Asian option numerically, it reduces computational cost and increases accuracy of the numerical solution. However, it may not always be possible to find a similarity solution [1] and derived one spatial dimensional PDEs are not suitable for barrier type of Asian option and local volatility model [13]. Also, combining other properties of other exotic option with Asian option will make problems more complicated. It is hard to find boundary conditions since one spatial dimensional PDE is not intuitive than two spatial dimensional PDE. Thus, there is necessity of good numerical algorithm to solve two spatial dimensional PDE in recent complicated financial market situation. We are going to adapt OSM and method of characteristic for one of spatial dimension which has no diffusion term to solve two spatial dimensional PDE. This concept was introduced by Duffy in [1] but was not described in detail. To show performance of the proposed algorithm, we will compare with MCS method.

Section 2 presents arithmetic Asian option and PDE derived by Ingersoll in [9]. Section 3 describes numerical solution algorithm using finite difference scheme and OSM with method of characteristic and presents a numerical algorithm to price arithmetic Asian option by using MCS method. Section 4 presents the computational results showing the performances of the MCS method and FDM using OSM, that is, comparative study. Conclusions are presented in Section 5.

2. Arithmetic Asian option PDE by Ingersoll

We adapt the classical Geometric Brownian Motion (GBM) for stock process $S(t)$:

$$dS(t) = rS(t)dt + \sigma S(t)dW(t) \tag{2.1}$$

where r is risk-neutral constant interest rate, σ is constant volatility and $dW(t)$ is standard Brownian Motion. $V(T, S(T), I(T))$ denote payoff function of Asian option where

$$I(t) = \int_0^t S(\theta)d\theta.$$

By definition, dynamics of variable I is

$$dI(t) = S(t)dt$$

which will be used for Itô's lemma. There must exist some function $v(t, S(t), I(t))$ such that

$$\begin{aligned} v(t, S(t), I(t)) &= \tilde{\mathbb{E}}\left[\frac{D(T)}{D(t)}V(T, S(T), I(T))|\mathcal{F}(t)\right] \\ &= \tilde{\mathbb{E}}\left[e^{-r(T-t)}V(T, S(T), I(T))|\mathcal{F}(t)\right] \end{aligned} \tag{2.2}$$

where $\tilde{\mathbb{E}}$ represent risk-neutral expectation and $D(t)$ denote discount factor. Since we assumed constant interest rate, discount factor is $D(t) = e^{-rt}$. Price function $v(t, S(t), I(t))$ can represent any strike type of Asian option. By property of risk-neutral pricing, process $D(t)v(t, S(t), I(t)) = e^{-rt}v(t, S(t), I(t))$ must be martingale. By applying Itô's lemma, dynamics of $D(t)v(t, S(t), I(t))$ becomes

$$\begin{aligned} d(e^{-rt}v(t, S(t), I(t))) &= e^{-rt}\left[-rvdt + v_t dt + v_S dS + v_I dI + \frac{1}{2}v_{SS}dSdS\right] \\ &= e^{-rt}\left[-rv + v_t + rSv_S + Sv_I + \frac{1}{2}\sigma^2 v_S S\right]dt + e^{-rt}\sigma S v_S d\widetilde{W}(t) \end{aligned}$$

where v_x denote differential of v on x-direction and $\widetilde{W}(t)$ is risk-neutral Brownian motion. For this process to be martingale, dt term of this process must be zero. Thus, we derived Ingersoll's 2-spatial dimensional PDE as follows

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} + S\frac{\partial V}{\partial I} - rV = 0. \tag{2.3}$$

To demonstrate following numerical algorithms, we set payoff function as fixed strike call

$$V(T, S(T), I(T))\Lambda(T) = (A(T) - K)^+. \tag{2.4}$$

Without loss of generality, other payoff function also can be adapted in those algorithms by changing some boundary conditions and initial conditions.

3. Numerical methods

In this section, we describe the numerical discretization of Eq. (2.3). We also present the operator-splitting algorithm in detail. Particularly, we are going to use method of characteristic for one of spatial dimension which does not have diffusion term.

3.1. **Finite difference discretization.**

Let $\mathcal{L}_{BS}$ be the operator

$$\mathcal{L}_{BS} = \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 v}{\partial s^2} + rs\frac{\partial v}{\partial s} + s\frac{\partial v}{\partial i} - rv$$

and τ be the remaining time to maturity such that $\tau = T - t$ where T is maturity. Then, the two spatial dimensional Ingersoll's equation can be rewritten as

$$\frac{\partial v}{\partial \tau} = \mathcal{L}_{BS}, \text{ for } (\tau, s, i) \in [0, T] \times \Omega,$$

Value of this derivative is defined on unbounded domain such that $\{(\tau, s, i) \mid \tau \in [0, T],\ s \geq 0,\ i \in \mathbb{R}\}$ as you can see in [14]. To compute price of option in computer, we truncate this domain as finite computational domain such that $\{(\tau, s, i) \mid \tau \in [0, T],\ 0 \leq s \leq M_s,\ 0 \leq i \leq M_i\}$, where M_s and M_i are large enough number. In general, M_s and M_i could be set as two or three times of strike price K. Then, error of the price driven by truncation becomes ignorable [15]. Note that setting of domain i is $\{i \mid 0 \leq i \leq M_i\}$. Although the original domain for i is $\{i \mid i \in \mathbb{R}\}$, we assumed that i is almost surely positive because I is defined as $I(t) = \int_0^t S(\theta) d\theta$ and classically defined GBM stock process $S(t)$ is positive on any time. We have Dirichlet boundary condition when $s = 0$ such that

$$v(\tau, 0, i) = e^{-r\tau} (\frac{i}{T} - K)^+$$

for $\tau \in [0, T]$ and $0 \leq i \leq M_i$. Also, for rest of three artificial boundaries, we have linear boundary condition [16, 17, 18, 19] such that

$$\frac{\partial^2}{\partial i^2} v(\tau, s, 0) = \frac{\partial^2}{\partial i^2} u(\tau, s, M_i) = \frac{\partial^2}{\partial s^2} u(\tau, M_s, i) = 0$$

for $\tau \in [0, T]$, $0 \leq s \leq M_s$ and $0 \leq i \leq M_i$.

Let N_s, N_i and N_τ denote the numbers of grid points for s-, i- and τ-directions, respectively. We are going to adapt uniform grid as $h = M_s/N_s = M_i = /N_i$ and $\Delta\tau = T/N_\tau$ which discretize the settled computational domain $[0, T] \times \Omega$ where $\Omega = (0, M_s) \times (0, M_i)$. Figure 1 illustrates the 2-dimensional uniform grid with a spatial step size h.

According to discretization, v_{jk}^n denote approximated numerical solution such that

$$v_{jk}^n = v(\tau^n, s_j, i_k) = v(n\Delta\tau, jh, kh)$$

for $n = 0, \ldots, N_\tau$, $j = 1, \ldots, N_s - 1$ and $k = 1, \ldots, N_i - 1$. For Dirichlet boundary condition when $s = 0$, numerical approximation in each time steps defined as

$$v_{0k}^n = v(n\Delta\tau, 0, kh) = e^{-rn\Delta\tau} (\frac{kh}{T} - K)^+.$$

Since other directions are used linear boundary condition and formula used linear boundary condition do not depend on time step, for all time steps, numerical approximation can be denoted as

$$v_{j0} = 2v_{j1} - v_{j2},\ v_{j,N_i} = 2v_{j,N_i-1} - v_{j,N_i-2}$$

for $j = 1, \cdots, N_s - 1$ and

$$v_{N_s,k} = 2v_{N_s-1,k} - v_{N_s-2,k}$$

for $k = 1, \cdots, N_i - 1$.

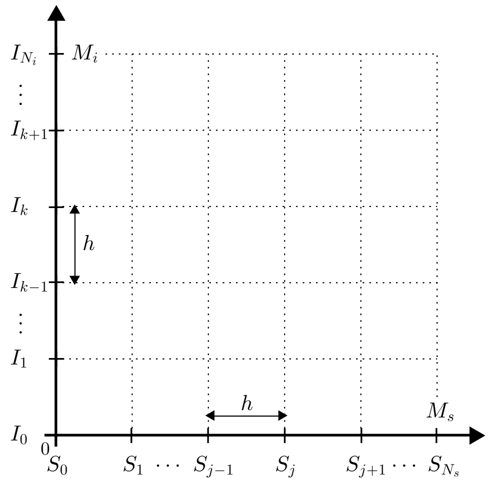


FIGURE 1. Schematic of uniform 2-dimensional grid with a spatial step size h.

3.2. **Operator-splitting method.** The concept of operator-splitting method is to transfer problem of multi-dimensional PDE to problem of multiple one-dimensional problems [1, 20]. In this case, we are going to use method of characteristic for i-direction whose solution could be represented as analytic solution by solving simple partial differential equation. Simple partial differential equation is

$$V_\tau - SV_I = 0. \tag{3.1}$$

We can find analytic solution by some work with Eq.(3.1) such that

$$V(S, I, \tau) = V(S, I(\tau - \tau^n), \tau^n) \text{ for } \tau \geq \tau^n$$

Verification of solution is as follows:

$$\begin{aligned} \text{Since} \quad & V_\tau = SV'(S, I + S(\tau - \tau^n), \tau^n) \text{ and } V_I = V'(S, I + S(\tau - \tau^n), \tau^n), \\ \text{hence} \quad & V_\tau - SV_I = SV' - SV' = 0. \end{aligned}$$

Adapting our discretized scheme to this solution, solution for each time step could be represented as

$$V(S, I, \tau^{n+1}) = V(S, I + S\Delta\tau, \tau^n). \tag{3.2}$$

Also, because solution could be in between points that we have discretized, we are going to use linear interpolation for computing this problem. In general, the basic operator-splitting scheme for the spatial two-dimensional Ingersoll's PDE as follows:

$$\frac{v_{jk}^{n+1} - v_{jk}^n}{\Delta\tau} = \mathcal{L}_{IG}^i v_{jk}^{n+\frac{1}{2}} + \mathcal{L}_{IG}^s v_{jk}^{n+1}, \tag{3.3}$$

where the operator $\mathcal{L}^i_{IG}$ and $\mathcal{L}^s_{IG}$ defined by

$$\mathcal{L}^i_{IG} v^{n+\frac{1}{2}}_{jk} = \frac{v^n_{j,k+1} - v^n_{j,k}}{h} s_j$$

$$\mathcal{L}^s_{IG} v^{n+1}_{jk} = \frac{(\sigma s_j)^2}{2} \frac{v^{n+1}_{j-1,k} - 2v^{n+1}_{jk} + v^{n+1}_{j+1,k}}{h^2} + r s_j \frac{v^{n+1}_{j+1,k} - v^{n+1}_{jk}}{h} + r v^{n+1}_{jk}.$$

The first step is using analytic solution in the i-direction. Next step is solving implicitly in the s-direction. We set temporary time step $n+\frac{1}{2}$ for compute OS-scheme which does not exist in real. This OSM reduce 2-dimensional problem into two 1-dimensional problem.

By Eq.(3.2), discrete solution is

$$v^{n+\frac{1}{2}}_{jk} = \frac{v^n_{j,k+1} - v^n_{jk}}{h} s_j \Delta\tau + v^n_{jk}. \tag{3.4}$$

This implies discrete difference i-direction operator using method of characteristic $\mathcal{L}^i_{IG}$ is defined by

$$\frac{v^{n+\frac{1}{2}}_{jk} - v^n_{jk}}{\Delta\tau} = \mathcal{L}^i_{IG} v^{n+\frac{1}{2}}_{jk}. \tag{3.5}$$

Also, implicitly approximating s-direction operator $\mathcal{L}^s_{IG}$ is defined by

$$\frac{v^{n+1}_{jk} - v^{n+\frac{1}{2}}_{jk}}{\Delta\tau} = \mathcal{L}^i_{IG} v^{n+1}_{jk}. \tag{3.6}$$

Note that sum of two Eqs.(3.5) and (3.6) is Eq.(3.3). Before iterate Algorithm's logic we set $v^0_{jk} = \Lambda(T)$. An algorithm of the OSM is as follows:

Algorithm OS

- *Step 1*

  Initialize $v^n_{0k} = e^{-rn\Delta\tau}(\frac{kh}{T} - K)^+$ in every time step.

- *Step 2*

  To describe as general form of OSM, Eq. (3.5) is used. However, to compute i-direction solution, Eq. (3.5) is rewritten as Eq.(3.4). For each j, we have

$$v^{n+\frac{1}{2}}_{jk} = \frac{v^n_{j,k+1} - v^n_{jk}}{h} s_j \Delta\tau + v^n_{jk} \tag{3.7}$$

  The first step of the OS method is then implemented in a loop over the s-direction:

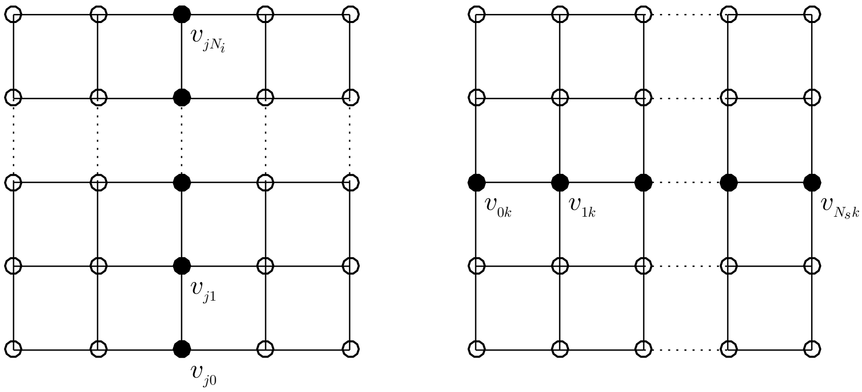
  for $j = 0 : N_s$
  for $k = 1 : N_i - 1$
  Solve $v^{n+\frac{1}{2}}_{jk}$ by Eq. (3.7) (see Fig.2(a))
  end

Use boundary condition

$$v_{j0}^{n+\frac{1}{2}} = 2v_{j1}^{n+\frac{1}{2}} - v_{j2}^{n+\frac{1}{2}},\ v_{j,N_i}^{n+\frac{1}{2}} = 2v_{j,N_i-1}^{n+\frac{1}{2}} - v_{j,N_i-2}^{n+\frac{1}{2}}$$

end

Note that $v_{0,k}^{n+\frac{1}{2}} = v_{0k}^n$ for $k = 0, \ldots, N_i$ because $s_0 = 0$. Thus, Step 2 do nothing for $v_{0,0:N_i}$.



(a) Step 2 (b) Step 3

FIGURE 2. Two steps of the OSM.

- *Step 3*

Unlike in *Step 1*, Eq. (3.6) is rewritten as follows:

$$\alpha_j v_{j-1,k}^{n+1} + \beta_j v_{jk}^{n+1} + \gamma_j v_{j+1,k}^{n+1} = f_{jk}, \tag{3.8}$$

where

$$\alpha_j = -\frac{1}{2}\frac{\sigma^2 s_j^2}{h^2},\ \beta_j = \frac{1}{\Delta\tau} + \frac{\sigma^2 s_j^2}{h^2} + \frac{r s_j}{h} + r,$$
$$\gamma_j = -\frac{1}{2}\frac{\sigma^2 s_j^2}{h^2} - \frac{r s_j}{h},\ \text{for } k = 1, ..., N_i - 1$$

and

$$f_{1k} = \frac{v_{1k}^{n+\frac{1}{2}}}{\Delta\tau} - \alpha_1 e^{-rn\Delta\tau}(\frac{kh}{T} - K)^+, \tag{3.9}$$

$$f_{jk} = \frac{v_{jk}^{n+\frac{1}{2}}}{\Delta\tau} \ \text{ for } j = 2, \ldots, N_s - 1. \tag{3.10}$$

Note that we adapt Dirichlet boundary condition to get the coefficients.

As with *Step 2*, *Step 3* is then operated in a loop over the i-direction:

for $\quad k = 0 : N_i$
  for $\quad j = 1 : N_y - 1$
    Set f_{jk} by Eq. (3.10)
  end
  Solve $A_s u^{n+1}_{1:N_s-1,k} = f_{1:N_s-1,k}$ by using Thomas algorithm (see Fig. 2(b))
end

Here A_s is tridiagonal matrix constructed from Eq. (3.8) with a Dirichlet boundary condition for $j = 0$ and a linear boundary condition for $j = N_s$

$$A_s = \begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 & 0 \\ 0 & \alpha_3 & \beta_3 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \beta_{N_y-1} & \gamma_{N_y-1} \\ 0 & 0 & 0 & \dots & \alpha_{N_y} - \gamma_{N_y} & \beta_{N_y} + 2\gamma_{N_y} \end{pmatrix}.$$

- *Step 4*

  Iterate Step 1 to 3 for all time step.

The information of OSM in this paper was based in [1]. For more information on OSM, please refer to [1].

3.3. **Monte Carlo simulation method.** To describe general Monte Carlo simulation, we also adapt Eq.(2.1). Integrate both side of Eq.(2.1) and rearrange by $S(t)$ equation becomes

$$S(t) = S(0)e^{(r-0.5\sigma^2)t+\sigma W(t)}.$$

Let Δt be small time increment and $S(n\Delta t)$ be denoted by S^n. Then, discretized stock process is defined as

$$S^{n+1} = S^n e^{(r-0.5\sigma^2)\Delta t+\sigma\sqrt{\Delta t}Z^n} \qquad \text{for} \quad 0 \le n \le \frac{T}{\Delta t} - 1 \tag{3.11}$$

where n is integer and random variable Z^n is independent and identically distributed (i.i.d) with standard Normal distribution which is denoted as $N(0,1)$. According to Eq.(3.11), we generate enough large path to satisfy consistency of estimation. (See Fig. 3.) Let M denote the number of iteration and Ω denote set of all parameters and sample space. Then,

$$\hat{v}(0, S(0), I(0)) = \frac{1}{M}e^{-rT}\sum_{i=1}^{M}\Lambda(T) \tag{3.12}$$

becomes good estimation of $v(0, S(0), I(0))$ by Eq.(2.2) where $\Lambda(T)$ represented as Eq.(2.4).

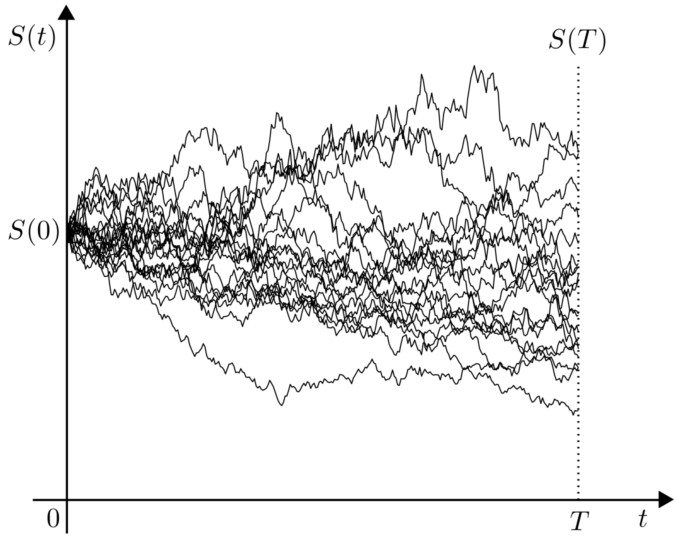


FIGURE 3. Schematic of path generation of MCS.

More information of Monte Carlo simulation method in finance is introduced in [2].

4. COMPUTATIONAL RESULTS

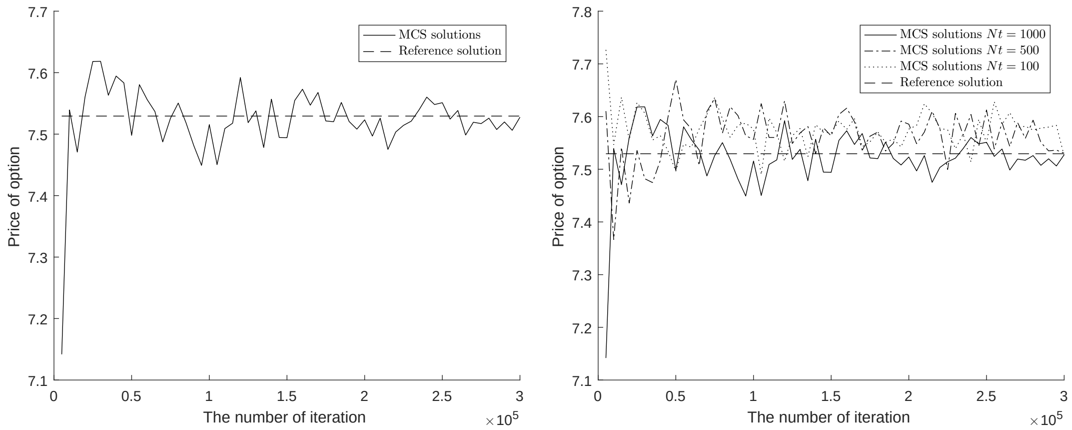
4.1. **Convergence test of FDM and MCS.** To demonstrate consistency of the numerical algorithms, we perform convergence tests. We set some parameters as $r = 0.03, S(0) = 100, K = 100, T = 1, \sigma = 0.3$. We test numerical scheme Eq.(3.3) with respect to h and $\Delta\tau$ for FDM. For truncated computational domain, we also set $M_s = M_i = 300$ and we are going to use these values unless otherwise specified. Table 1 show the convergence of Asian fixed strike type call option prices at $(S(0), I(0), 0)$ as we refine h and $\Delta\tau$. As you see, tendency of prices flow with respect to h and $\Delta\tau$ seem to be converges to a certain value.

TABLE 1. Convergence test for European call option values by FDM.

Case	$h = 4$	$h = 2$	$h = 1$	$h = 0.5$	$h = 0.25$	$h = 0.125$
$\Delta\tau = 1/90$	10.250442	8.496569	7.677030	7.610064	7.579022	7.569500
$\Delta\tau = 1/180$	10.666462	9.004364	8.024811	7.598022	7.564093	7.548436
$\Delta\tau = 1/360$	10.867678	9.246414	8.299454	7.776086	7.558052	7.540982
$\Delta\tau = 1/720$	10.966702	9.364765	8.432927	7.919678	7.648182	7.537956

Since Arithmetic Asian option does not have analytic solution, we need to set reference solution. We set $V(S(0), I(0), 0) = 7.529395$ as reference solution by FDM very slight grid such as $h = 0.0625$ and $\Delta\tau = 1/720$.

For MCS, we calculate estimation Eq.(3.12) with respect to the number of iteration. Let N_m denote the number of iteration. Figure 4 shows convergence of MCS method. As N_m increases, numerical solution of MCS also seem to converge to a certain value. (See figure 4 (a).) We compute prices by 5000 interval of iteration. Figure 4 (b) shows convergence with respect to Δt. To approximate integration of stock process, that is, $\int_0^T S(t)dt \approx \sum_0^{T/\Delta t} S^n \Delta t$, we set $\Delta t = 1/10000$ unless otherwise specified so that the error of numerical integration becomes negligible. Table 2 represent numerical solutions by 50000 interval of iteration.



(a) Convergence of MCS (b) Convergence with respect to Δt

FIGURE 4. Convergence test of MCS.

TABLE 2. Convergence test for European call option values by MCS.

N_m	50000	100000	150000	200000	250000	300000
Price	7.498124	7.515509	7.494195	7.523239	7.551128	7.527592

Our test could be enough evidence that both FDM and MCS converges to a certain value.

In this section, we show the performance of the proposed numerical algorithms by numerical experiments. All performance were computed on a 2.7 GHz Intel PC 8 GB of RAM loaded with MATLAB 2016a [21]. We consider three case with respect to money-ness such as in the money, at the money and out of the money. Table 3 represent the parameters we set. In this section, we also set reference solution by FDM with fine grid size such as $h = 0.0625$ and $\Delta\tau = 1/720$. For at the money price, we use 7.529395 which is used for convergence test. Other values 21.423097 and 1.743159 are in the money and out of the money reference solution each.

TABLE 3. Parameters

risk-neutral interest rate(r)	0.03	dividend(q)	0
maturity(T)	1	volatility(σ)	0.3
initial stock price $S(0)$	100	in the money strike(K_1)	80
at the money strike(K_2)	100	out of the money strike(K_3)	120

4.2. **Computational results of Finite Difference Method.** We perform results of FDM with respect to h and $\Delta\tau$. In general, spatial grid h could be set as 0.5 or 1 for spatially one-dimensional PDE. This value could be considered as enough small value. However, since proposed algorithm is spatially two-dimensional PDE, results shows that spatial grid size should be smaller. Thus, we set spatial grid h as 0.25 unless otherwise specified. Table 4, 5 and 6 shows numerical results.

TABLE 4. In the money results of FDM with $h = 0.25$ and Iteration number ($1/\Delta\tau$)

Strike	Iteration Number	Reference Price	Numerical Price	Error (%)	Computational Cost
80	90	21.423097	21.459686	0.171	0.403min
	180		21.446979	0.111	0.812min
	360		21.441516	0.086	1.574min
	720		21.489467	0.310	3.145min

TABLE 5. At the money results of FDM with $h = 0.25$ and Iteration number ($1/\Delta\tau$)

Strike	Iteration Number	Reference Price	Numerical Price	Error (%)	Computational Cost
100	90	7.529395	7.579022	0.659	0.400min
	180		7.564093	0.461	0.780min
	360		7.558052	0.381	1.577min
	720		7.648182	1.578	3.086min

TABLE 6. Out of the money results of FDM with $h = 0.25$ and Iteration number ($1/\Delta\tau$)

Strike	Iteration Number	Reference Price	Numerical Price	Error (%)	Computational Cost
120	90	1.743159	1.784897	2.394	0.391min
	180		1.768427	1.450	0.793min
	360		1.770094	1.545	1.559min
	720		1.792457	2.828	3.091min

As you see the tables, numerical solutions seem to converge to reference solution except the case that $\Delta\tau = 1/720$. We could interpret this results as time error become negligible against spatial error by the fact that FDM has spatial error $O(h^2)$ and time error $O(\Delta\tau)$

4.3. **Computational results of Monte Carlo Simulation.** Table 7, 8 and 9 are numerical results of MCS. We formed the following tables like FDM solution tables to compare results.

TABLE 7. In the money results of MCS

Strike	Iteration Number	Reference Price	Numerical Price	Error (%)	Computational Cost
80	5000	21.423097	21.275702	0.688	0.019min
	10000		21.734840	1.455	0.047min
	50000		21.347946	0.351	2.272min
	100000		21.474813	0.241	11.461min

TABLE 8. At the money results of MCS

Strike	Iteration Number	Reference Price	Numerical Price	Error (%)	Computational Cost
100	5000	7.529395	7.284143	3.257	0.021min
	10000		7.638652	1.451	0.043min
	50000		7.546381	0.226	2.210min
	100000		7.517309	0.161	10.942min

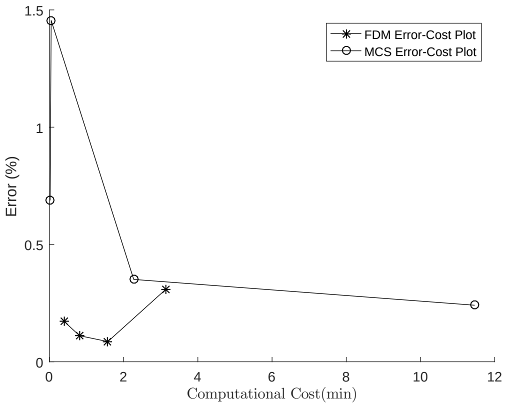
TABLE 9. Out of the money results of MCS

Strike	Iteration Number	Reference Price	Numerical Price	Error (%)	Computational Cost
120	5000	1.743159	1.535304	11.924	0.019min
	10000		1.635084	6.200	0.041min
	50000		1.715868	1.566	2.245min
	100000		1.737488	0.325	11.147min

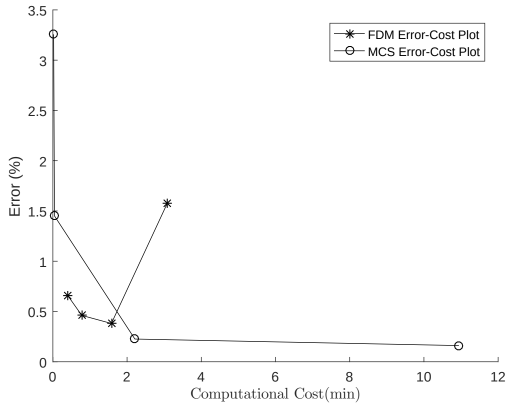
As you see the tables, solutions of MCS seem to converge to reference solution. However, MCS solution could not monotonously converge to reference solution ,whereas FDM solution tend to go to reference solution monotonously as N_m increase except what we have mentioned previous subsection. Figure 5 shows this fact. In practical field, the number of simulation N_m is set as 60000 which is assumed to be enough number to get consistent estimator with various technique such as quasi random number [22], Brownian bridge [23] and variance reduction [24, 25]. In this paper, we set the maximum number of iteration as 100000 without any technique.

5. CONCLUSIONS

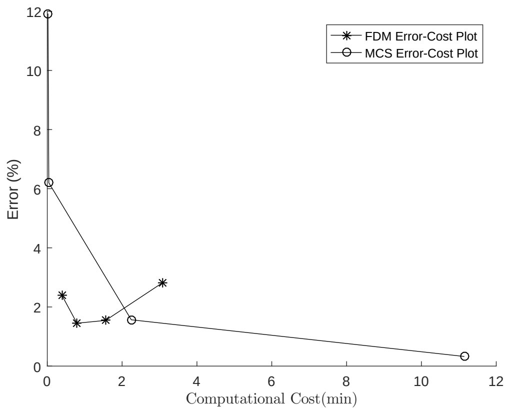
In this paper, we presented a numerical algorithm for arithmetic Asian option by using the OSM and method of characteristic with 2-dimensional PDE. We construct the algorithm to approximate the value of Asian option which does not have analytic solution. We modeled Ingersoll's spatially 2-dimensional PDE by adapting FDM with OSM and method of characteristic. We describe a detailed numerical algorithm and computational results illustrating



(a) Error-Cost Plot in case $K = 80$



(b) Error-Cost Plot in case $K = 100$



(c) Error-Cost Plot in case $K = 120$

FIGURE 5. Error-Cost Plot.

performance of proposed algorithms. As we mentioned previously, there are many techniques to evolve MCS. There are also various techniques to evolve FDM such as adapting adaptive schemes and high order schemes, various ways to deal with boundary condition to reduce time and etc. In this paper, we compared pure FDM and MCS so that we introduce simple basic algorithm to be understood easily and be adapted easily to various ways which evolve algorithms more efficient. Fig. 5 (a) show that FDM is superior than MCS. Fig. 5 (b) and (c) see also our proposed FDM algorithm is good except extraordinary price which we have mentioned in section of results of FDM. Indeed, our proposed FDM algorithm is competitive even though we consider extraordinary price made by fact that spatial error covers time error.

ACKNOWLEDGMENTS

The first author (Jian Wang) was supported by the Brain Korea 21 Project from the Ministry of Education of Korea. This paper was based on master's thesis of the corresponding author (Changwoo Yoo).

REFERENCES

[1] Duffy, Daniel J. Finite Difference methods in financial engineering: a Partial Differential Equation approach. John Wiley & Sons, 2013.
[2] GLASSERMAN, Paul. Monte Carlo methods in financial engineering. Springer Science & Business Media, 2013.
[3] SEYDEL, Rudiger; SEYDEL, Rudiger. Tools for computational finance. Berlin: Springer, 2006.
[4] TURNBULL, Stuart M.; WAKEMAN, Lee Macdonald. A quick algorithm for pricing European average options. Journal of financial and quantitative analysis, 1991, 26.3: 377–389.
[5] Vorst, Ton. VORST, Ton. Prices and hedge ratios of average exchange rate options. International Review of Financial Analysis, 1992, 1.3: 179–193.
[6] LEVY, Edmond. Pricing European average rate currency options. Journal of International Money and Finance, 1992, 11.5: 474–491.
[7] LEVY, Edmond; TURNBULL, Stuart. Average intelligence. Risk, 1992, 5.2: 53–57.
[8] CEN, Zhongdi; XU, Aimin; LE, Anbo. A hybrid finite difference scheme for pricing Asian options. Applied Mathematics and Computation, 2015, 252: 229–239.
[9] INGERSOLL, Jonathan E. Theory of financial decision making. Rowman & Littlefield, 1987.
[10] BLACK, Fischer; SCHOLES, Myron. The pricing of options and corporate liabilities. Journal of political economy, 1973.
[11] WILMOTT, Paul; DEWYNNE, Jeff; HOWISON, Sam. Option pricing: mathematical models and computation. Oxford financial press, 1993.
[12] ROGERS, L. Chris G.; SHI, Zo. The value of an Asian option. Journal of Applied Probability, 1995, 32.4: 1077–1088.
[13] FOSCHI, Paolo; PAGLIARANI, Stefano; PASCUCCI, Andrea. Approximations for Asian options in local volatility models. Journal of Computational and Applied Mathematics, 2013, 237.1: 442–459.
[14] S. Shreve, Stochastic calculus for finance II: Continuous-time models, Springer Science and Business Media, Pennsylvania (2004)
[15] KANGRO, Raul; NICOLAIDES, Roy. Far Field Boundary Conditions for Black–Scholes Equations. SIAM Journal on Numerical Analysis, 2000, 38.4: 1357–1368.
[16] OOSTERLEE, Cornelis W. On multigrid for linear complementarity problems with application to American-style options. Electronic Transactions on Numerical Analysis, 2003, 15.1: 165–185.
[17] PERSSON, Jonas; VON PERSSON, Lina. Pricing European multi-asset options using a space-time adaptive FD-method. Computing and Visualization in Science, 2007, 10.4: 173–183.
[18] TAVELLA, Domingo; RANDALL, Curt. Pricing financial instruments: The finite difference method. John Wiley & Sons, 2000.
[19] ZVAN, Robert; VETZAL, Kenneth R.; FORSYTH, Peter A. PDE methods for pricing barrier options. Journal of Economic Dynamics and Control, 2000, 24.11-12: 1563–1590.
[20] IKONEN, Samuli; TOIVANEN, Jari. Operator splitting methods for American option pricing. Applied mathematics letters, 2004, 17.7: 809–814.
[21] Using MATLAB, The MathWorks Inc., Natick, MA, 1998, http://www.mathworks.com/.
[22] XU, Yongjia; LAI, Yongzeng; YAO, Haixiang. Efficient simulation of Greeks of multiasset European and Asian style options by Malliavin calculus and quasi-Monte Carlo methods. Applied mathematics and computation, 2014, 236: 493–511.

[23] UEKI, Taro. Brownian bridge and statistical error estimation in Monte Carlo reactor calculation. Journal of Nuclear Science and Technology, 2013, 50.8: 762–773.

[24] JACOBSON, Sheldon H. Variance and bias reduction techniques for the harmonic gradient estimator1. Applied Mathematics and Computation, 1993, 55.2-3: 153–186.

[25] DANG, Duy-Minh; JACKSON, Kenneth R.; MOHAMMADI, Mohammadreza. Dimension and variance reduction for Monte Carlo methods for high-dimensional models in finance. Applied Mathematical Finance, 2015, 22.6: 522–552.