axioms

MDPI

Article

Efficient BEM-Based Algorithm for Pricing Floating Strike Asian Barrier Options (with MATLAB® Code)

Alessandra Aimi [1,*,†,‡], **Lorenzo Diazzi** [2,‡] **and Chiara Guardasoni** [3,‡]

[1] Department of Mathematical Physical and Computer Sciences, University of Parma, Parma 43124, Italy
[2] Department of Physics Informatics and Mathematics, University of Modena and Reggio Emilia, Modena 41125, Italy; lorenzo.diazzi@unimore.it
[3] Department of Mathematical Physical and Computer Sciences, University of Parma, Parma 43124, Italy; chiara.guardasoni@unipr.it
[*] Correspondence: alessandra.aimi@unipr.it; Tel.: +39-0521-906-944
[†] Current address: Parco Area delle Scienze 53/A, 43124 Parma, Italy.
[‡] Members of the INdAM-GNCS Research Group. These authors contributed equally to this work.

Received: 11 May 2018; Accepted: 12 June 2018; Published: 15 June 2018

Abstract: This paper aims to illustrate how SABO (Semi-Analytical method for Barrier Option pricing) is easily applicable for pricing floating strike Asian barrier options with a continuous geometric average. Recently, this method has been applied in the Black–Scholes framework to European vanilla barrier options with constant and time-dependent parameters or barriers and to geometric Asian barrier options with a fixed strike price. The greater efficiency of SABO with respect to classical finite difference methods is clearly evident in numerical simulations. For the first time, a user-friendly MATLAB® code is made available here.

Keywords: boundary element method; finite difference method; floating strike Asian options; continuous geometric average; barrier options

1. Introduction

The pricing of continuously-monitored Asian options is a relevant task from both a mathematical and a financial point of view.

Asian options are quite common derivatives because they provide protection against strong price fluctuations in volatile markets and reduce the possibilities of price manipulations. The payoff of an Asian option depends on the average price of the underlying asset that is less volatile than the asset price itself. In general, Asian options are hence less valuable than their vanilla European counterparts because an option on a lower volatility asset is worth less.

On the other hand, it is more difficult to deal with Asian options than vanilla options because their price depends on the average value assumed by the underlying asset during the option's life, requiring some mathematical effort in order to describe the dynamics of the average under consideration.

In this paper, Asian options are equipped with a continuously-monitored geometric average [1]. Asian options evaluated with the geometric mean, although not common among practitioners, give some information also about the evaluation of Asian options with the arithmetic mean [2]. From a theoretical point of view, the method illustrated in the paper is extensible to arithmetic Asian options, as well, with slight modification, but from the numerical point of view, there are several problems that we plan to investigate in the near future. Defining the stochastic process

$$A_t := \int_0^t \log(S_\tau) d\tau \tag{1}$$

then the geometric average is defined as $\exp\left(\frac{A_t}{t}\right)$. When A and S are written with subscripts (A_t and S_t), they are intended as stochastic processes; otherwise, they are considered independent variables in the differential analysis context. The differential problem that describes the price evolution of this option is:

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} + \log(S) \frac{\partial V}{\partial A} - rV = 0\,, &\qquad S \in \mathbb{R}^+,\ A \in \mathbb{R},\ t \in [0, T) &(2)\\ V(S, A, T) \quad \text{assigned}\,, &\qquad S \in \mathbb{R}^+,\ A \in \mathbb{R}. &(3) \end{aligned}$$

Wanting to provide a further protection against excessive fluctuations of the strike price, it is possible to apply barriers in the option contract; for example, knock-out barriers make the option cease to exist if the underlying asset reaches a barrier during the life of the option. The model analyzed in this paper concerns an Asian option with an up-and-out barrier at $S = B$ and a floating strike payoff, i.e.,

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} + \log(S) \frac{\partial V}{\partial A} - rV = 0\,, &\qquad S \in (0, B),\ A \in \mathbb{R},\ t \in [0, T) &(4)\\ V(S, A, T) \quad \text{assigned}\,, &\qquad S \in (0, B),\ A \in \mathbb{R} &(5)\\ V(B, A, t) = 0\,, &\qquad A \in \mathbb{R},\ t \in [0, T) &(6)\\ \text{asymptotic conditions of vanilla option}\,, &\qquad \{(S, A) : S = 0 \vee A \to -\infty \vee A \to +\infty\}, &(7) \end{aligned}$$

with the final condition:

$$\begin{aligned} &\text{call} \qquad V(S, A, T) = \max\left(S - \exp\left(\frac{A}{T}\right), 0\right) \qquad \text{or} &(8)\\ &\text{put} \qquad V(S, A, T) = \max\left(\exp\left(\frac{A}{T}\right) - S, 0\right). &(9) \end{aligned}$$

The problem (2)–(3) of pricing a floating strike Asian option with a continuous geometric average and without a barrier has a closed-form solution in the domain $(S, A, t) \in \mathbb{R}^+ \times \mathbb{R} \times [0, T)$ that can be formulated either as the payoff expected value (also known as the Feynman–Kac formula):

$$V(S, A, t) = \int_{-\infty}^{+\infty} \int_0^B V(\widetilde{S}, \widetilde{A}, T) G(S, A, t; \widetilde{S}, \widetilde{A}, T) d\widetilde{S}\, d\widetilde{A}\,, \tag{10}$$

with the transition probability density function G associated with the differential operator defined in Equation (4), which is known to be:

$$\begin{aligned} G(S, A, t; \widetilde{S}, \widetilde{A}, \widetilde{t}) \;=\; & \frac{\sqrt{3}H[\widetilde{t}-t]}{\pi\sigma^2(\widetilde{t}-t)^2} \exp\left\{ -\frac{2}{\sigma^2(\widetilde{t}-t)} \log^2\left(\frac{S}{\widetilde{S}}\right) + \frac{6}{\sigma^2(\widetilde{t}-t)^2} \log\left(\frac{S}{\widetilde{S}}\right) \left(A - \widetilde{A} + (\widetilde{t} - t)\log(S)\right) \right.\\ & \left. -\; \frac{6}{\sigma^2(\widetilde{t}-t)^3} \left(A - \widetilde{A} + (\widetilde{t} - t)\log(S)\right)^2 - \left(\frac{2r+\sigma^2}{2\sqrt{2}\sigma}\right)^2 (\widetilde{t} - t) \right\} \left(\frac{\widetilde{S}}{S}\right)^{\frac{2r-\sigma^2}{2\sigma^2}} \frac{1}{\widetilde{S}}\,, \end{aligned} \tag{11}$$

or (see [3]) through the formula:

$$\begin{aligned} V_{\text{call}}(S, A, t) &= S\mathcal{N}[d] - e^{\frac{A}{T}} S^{\frac{T-t}{T}} e^q \mathcal{N}\left[d - \frac{\sigma}{T}\sqrt{\frac{T^3 - t^3}{3}}\right]\\ d &= \frac{t(\log(S) - \frac{A}{t}) + (r + \frac{\sigma^2}{2})\frac{T^2-t^2}{2}}{\sigma\sqrt{\frac{T^3-t^3}{3}}}\\ q &= \frac{(t - T)[6Tr(t + T) + (T - t)(2t + T)\sigma^2]}{12T^2} \end{aligned} \tag{12}$$

for the call option, where $\mathcal{N}[\cdot]$ is the normal cumulative distribution function, and eventually, the put-call parity:

$$V_{\text{call}}(S,A,t) - V_{\text{put}}(S,A,t) = S - e^{\frac{A}{T}} S^{\frac{T-t}{T}} e^{q}\,. \tag{13}$$

Instead, when applying barriers, no closed-formulas are available. In this context, SABO is a Semi-Analytical method conceived of for the pricing of Barrier Options, and its milestones are resumed in Section 4.1. It is quite a general method, applicable also to fixed strike payoffs [4,5], put options [6], time-dependent parameters [7] and double barriers [8].

SABO is compared here with two Finite Difference (FD) methods chosen among the wide class of numerical methods at our disposal [9]. Equation (4) is proven to be hypo-elliptic [10–12], a property that guarantees a smooth solution and should benefit from approximations based on Taylor expansions. Anyway, SABO appears to be certainly more efficient looking at the results below.

2. Results

We have performed several simulations related to the pricing of a geometric call Asian option with a floating strike price and with an up-and-out barrier as modeled by the differential problem (4)–(8). Numerical results, some of which are displayed in the following, have been obtained by the MATLAB® codes implementing the algorithms of SABO, FD1 and FD2 described in Sections 4.1, 4.2.1 and 4.2.2, respectively. In Section 5 the SABO code is provided.

Example 1. *In this example, we use the finance parameters displayed in Table 1. The floating strike call option with an up-and-out barrier at* $S = 150 =: B$ *is evaluated at* $t = 0$ *and* $A = 0$*, truncating the A-domain at* $A_{min} = 0$ *and* $A_{max} = 5$*, in accordance with (20) and either (19) or (27).*

The approximation by SABO is obtained setting the parameters described in Section 4.1 $N_t = N_A = 20$*, and the option value is displayed in Figure 1 as a function of S.*

The convergence at $S = 100, 120, 140, 148$ *is shown in Table 2 refining the mesh: at each level, parameters* N_t *and* N_A *are doubled.*

The approximation by FD1 is obtained setting the discretization parameters (defined in (17), (18) and (30)) $\Delta t = 10^{-3}/2^k$, $\Delta A = 10^{-2}/2^k$ *and, for the discretization of the asset domain* $(0, B)$*, either* $\Delta S = 2$ *or* $\Delta S = 1$*. The related results are reported in Tables 3 and 4.*

The approximation by FD2 is obtained setting the discretization parameters $\Delta t = \Delta A = 0.01/2^k$ *and either* $\Delta S = 2$ *or* $\Delta S = 1$*. The results are displayed in Tables 5 and 6.*

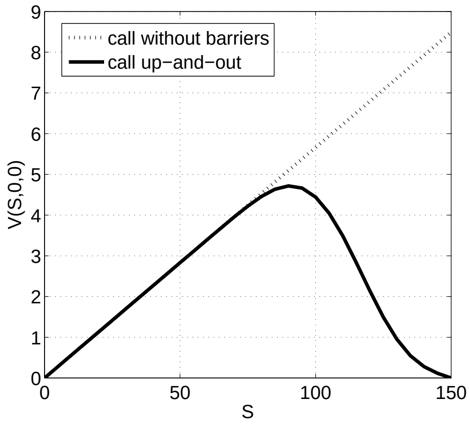

Figure 1. Semi-Analytical method for Barrier Option (SABO) approximation of a call up-and-out geometric Asian option with a floating strike and the data in Table 1.

Table 1. Floating strike up-and-out call option data.

B	T	r	σ
150	1	0.035	0.2

Table 2. $V(S, 0, 0)$ evaluated by SABO at $S = 100, 120, 140, 148$.

$N_t = N_A$	$S = 100$	$S = 120$	$S = 140$	$S = 148$	Elapsed Time (s)
20	4.4059	2.1078	0.2771	0.0441	9.0×10^0
40	4.4339	2.2728	0.3814	0.0632	3.1×10^1
80	4.4335	2.3506	0.4683	0.0807	1.3×10^2
160	4.4334	2.3600	0.4854	0.0843	5.1×10^2
320	4.4333	2.3616	0.4882	0.0849	2.2×10^3
640	4.4333	2.3619	0.4887	0.0850	1.0×10^4

Table 3. $V(S, 0, 0)$ evaluated by FD1at $S = 100, 120, 140, 148$, with $\Delta S = 2$.

k	$S = 100$	$S = 120$	$S = 140$	$S = 148$	Elapsed Time (s)
1	5.8385	3.6788	0.9933	0.1834	6.9×10^0
2	5.2000	3.0951	0.7729	0.1405	3.3×10^1
3	4.8342	2.7511	0.6412	0.1148	1.5×10^2
4	4.6360	2.5610	0.5673	0.1004	5.3×10^2
5	4.5324	2.4605	0.5278	0.0926	2.1×10^3
6	4.4794	2.4087	0.5073	0.0886	8.5×10^3
7	4.4526	2.3824	0.4968	0.0866	4.1×10^4

Table 4. $V(S, 0, 0)$ evaluated by FD1 at $S = 100, 120, 140, 148$, with $\Delta S = 1$.

k	$S = 100$	$S = 120$	$S = 140$	$S = 148$	Elapsed Time (s)
1	5.8427	3.6815	0.9946	0.1837	1.9×10^1
2	5.2047	3.0985	0.7744	0.1408	7.7×10^1
3	4.8392	2.7550	0.6429	0.1151	2.8×10^2
4	4.6413	2.5651	0.5691	0.1007	1.1×10^3
5	4.5380	2.4648	0.5296	0.0930	4.4×10^3
6	4.4551	2.4131	0.5092	0.0890	2.2×10^4
7	4.4583	2.3869	0.4988	0.0870	9.5×10^4

Table 5. $V(S, 0, 0)$ evaluated by FD2 at $S = 100, 120, 140, 148$, with $\Delta S = 2$.

k	$S = 100$	$S = 120$	$S = 140$	$S = 148$	Elapsed Time (s)
1	4.6607	2.5250	0.5016	0.0853	9.8×10^0
2	4.4796	2.3874	0.4806	0.0826	3.7×10^1
3	4.4394	2.3640	0.4845	0.0840	1.5×10^2
4	4.4296	2.3587	0.4861	0.0844	6.0×10^2
5	4.4269	2.3570	0.4864	0.0845	2.4×10^3
6	4.4261	2.3564	0.4864	0.0845	1.0×10^4

Table 6. $V(S,0,0)$ evaluated by FD2 at $S = 100, 120, 140, 148$, with $\Delta S = 1$.

k	$S = 100$	$S = 120$	$S = 140$	$S = 148$	Elapsed Time (s)
1	4.6657	2.5287	0.5034	0.0857	2.8×10^1
2	4.4853	2.3918	0.4826	0.0831	1.2×10^2
3	4.4452	2.3685	0.4865	0.0844	4.5×10^2
4	4.4354	2.3632	0.4881	0.0848	1.9×10^3
5	4.4327	2.3616	0.4884	0.0849	7.6×10^3
6	4.4318	2.3609	0.4884	0.0849	4.8×10^4

Example 2. *In this example, we use the finance parameters displayed in Table 7 and volatility equal to* $\sigma = 0.2, 0.3, 0.4$. *The floating strike call option with an up-and-out barrier at* $S = 115 =: B$ *is evaluated at* $t = 0$ *and* $A = 0$, *truncating the A-domain at* $A_{min} = 0$ *and* $A_{max} = 5$, *in accordance with* (20) *and* (19).

The approximation by SABO is obtained setting the parameters described in Section 4.1 $N_t = N_A = 20$ *and the option value is displayed in Figure 2 as a function of S for the different values of* σ *(continuous lines), in comparison with the corresponding prices of Asian options without barriers (dotted lines).*

Table 7. Floating strike up-and-out call option data.

B	T	r
115	1	0.08

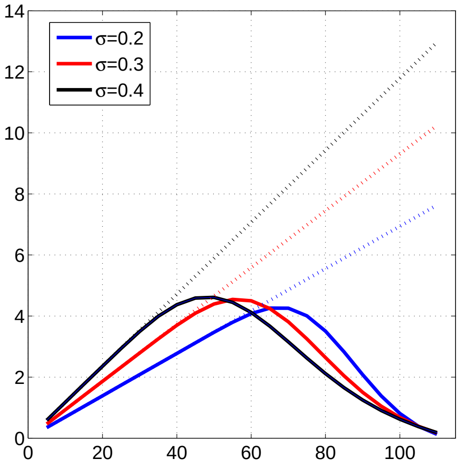

Figure 2. SABO approximation of a call up-and-out geometric Asian option with a floating strike, the data in Table 7 and various values of σ (continuous lines) compared with the corresponding prices of Asian options without barriers (dotted lines).

3. Discussion

Looking at Example 1, the values of a call option with an up-and-out barrier obtained by SABO and displayed in Figure 1 show that the solution, as expected, assumes lower values than the analogous option without barriers whose closed formula is (12) or that can be computed through the evaluation of the payoff expected value (10). The same behavior is recovered by the two proposed FD methods (FD1 and FD2).

Talking about efficiency and convergence, we have to look at the stabilization of the digits in Tables 2–6 where the option values at $S = 100, 120, 140, 148$ are written.

SABO, the results of which are written in Table 2, appears to be faster than the FD methods: doubling parameters N_t and N_A, the CPU time for computation quadruples, but one more digit of accuracy is achieved. The convergence is slower near the barrier because there, the barrier option value is more different from the option value without barriers: the option value without the barrier can be quasi-exactly computed by the Feynman–Kac Formula (10), and therefore, the approximation error introduced by SABO solving the boundary integral Equation (15) related to the barrier case is more involved in representation Formula (14) as the asset nears the barrier (look at (26)).

Comparing Tables 3 and 4, with an analogous computational time, SABO appears much more accurate and therefore efficient than FD1. Furthermore, note that FD1 is still sensitive to the mesh refinement in the S-domain: to halve ΔS means a significant variation in values of V together with a big increase of the computational costs.

Analyzing Tables 5 and 6, we observe that FD2 has a superior accuracy compared to FD1 due to its higher order of consistency: approximations of derivatives in the t and A variables are both of second order. Anyway, FD2 is less efficient than SABO, and the coarseness of the S-grid still significantly affects its results. Refinements in the S-grid would result in much longer computational times, no longer comparable with those of SABO.

Looking at Example 2, SABO maintains its robustness varying the volatility values. The solutions displayed in Figure 2 show the property of smoothness proven in [11]. The increase in volatility causes the expected increase in the value of vanilla options, but on the contrary, it implies a diminishing of barrier option values near the barrier.

4. Methods

4.1. SABO

SABO is a Semi-Analytical method for the pricing of Barrier Options. The foundation on which it is based is the integral representation of the solution of the modeling differential problem based on the knowledge of the transition probability density function.

For a problem like that in (4)–(7), it is proven in [4] that the integral representation formula is: $\forall(S, A, t) \in (0, B) \times \mathbb{R} \times [0, T)$

$$V(S,A,t) = \int_{-\infty}^{+\infty}\int_0^B V(\widetilde{S},\widetilde{A},T)G(S,A,t;\widetilde{S},\widetilde{A},T)d\widetilde{S}\,d\widetilde{A} + \int_t^T\int_{-\infty}^{+\infty}\frac{\sigma^2}{2}B^2\frac{\partial V}{\partial\widetilde{S}}(B,\widetilde{A},\widetilde{t})G(S,A,t;B,\widetilde{A},\widetilde{t})d\widetilde{A}\,d\widetilde{t} \qquad (14)$$

where the transition probability density function $G(S, A, t; B, \widetilde{A}, \widetilde{t})$ is defined in (11) and associated with the differential operator defined in Equation (4).

Note that in (14), both $V(S, A, t)$ and $\frac{\partial V}{\partial \widetilde{S}}(B, \widetilde{A}, \widetilde{t})$ are unknown, but, at $S = B$, the Boundary Condition (6) can be applied giving rise to the Volterra integral equation of the first kind:

$$0 = V(B,A,t) = \int_{-\infty}^{+\infty}\int_0^B V(\widetilde{S},\widetilde{A},T)G(B,A,t;\widetilde{S},\widetilde{A},T)d\widetilde{S}\,d\widetilde{A} + \int_t^T\int_{-\infty}^{+\infty}\frac{\sigma^2}{2}B^2\frac{\partial V}{\partial\widetilde{S}}(B,\widetilde{A},\widetilde{t})G(B,A,t;B,\widetilde{A},\widetilde{t})d\widetilde{A}\,d\widetilde{t} \qquad (15)$$

in the unknown $\frac{\partial V}{\partial \widetilde{S}}(B, \widetilde{A}, \widetilde{t})$.

The unknown is approximated by:

$$\frac{\partial V}{\partial \widetilde{S}}(B, \widetilde{A}, \widetilde{t}) \approx \sum_{k=1}^{N_t}\sum_{h=1}^{N_A} \alpha_h^{(k)} \psi_h(\widetilde{A})\varphi_k(\widetilde{t}) \qquad (16)$$

having defined piecewise constant basis functions:

$$\varphi_k(\widetilde{t}) := H[\widetilde{t} - t_{k-1}] - H[\widetilde{t} - t_k], \quad k = 1, \ldots, N_t\,,$$

on a uniform time grid:

$$t_k := k\Delta t, \quad k = 0, \ldots, N_t, \qquad \Delta t := \frac{T}{N_t}, \quad N_t \in \mathbb{N}^+\,, \qquad (17)$$

and piecewise constant basis functions:

$$\psi_h(\widetilde{A}) := H[\widetilde{A} - A_{h-1}] - H[\widetilde{A} - A_h], \quad h = 1, \ldots, N_A\,,$$

on a uniform A-grid over the truncated A-domain $[A_{\min}, A_{\max}]$:

$$A_h := A_{\min} + h\Delta A, \quad h = 0, \ldots, N_A, \qquad \Delta A := \frac{A_{\max} - A_{\min}}{N_A}, \quad N_A \in \mathbb{N}^+\,. \tag{18}$$

A careful choice of $A_{\min}$ and $A_{\max}$ has to be performed in such a way that the double integral:

$$\int_t^T \int_{-\infty}^{+\infty} G(B, A, t; B, \widetilde{A}, \widetilde{t}) d\widetilde{A}\, d\widetilde{t}$$

is rightly approximated by:

$$\int_t^T \int_{A_{\min}}^{A_{\max}} G(B, A, t; B, \widetilde{A}, \widetilde{t}) d\widetilde{A}\, d\widetilde{t}.$$

By the analysis developed in [5], this results in seeking for $A_{\max}$ such that:

$$-\int_t^T \frac{\exp\left(-\frac{(\widetilde{t}-t)(2r+\sigma^2)^2}{8\sigma^2}\right)}{2B\sigma\sqrt{2\pi(\widetilde{t}-t)}} \mathrm{Erf}\left[\frac{\sqrt{6}(A - A_{\max} + (\widetilde{t}-t)\log(B))}{(\widetilde{t}-t)^{3/2}\sigma}\right] d\widetilde{t} = \frac{\mathrm{Erf}\left[\frac{(2r+\sigma^2)\sqrt{T-t}}{2\sigma\sqrt{2}}\right]}{B(2r+\sigma^2)} \tag{19}$$

and $A_{\min}$ such that:

$$\int_t^T \frac{\exp\left(-\frac{(\widetilde{t}-t)(2r+\sigma^2)^2}{8\sigma^2}\right)}{2B\sigma\sqrt{2\pi(\widetilde{t}-t)}} \mathrm{Erf}\left[\frac{\sqrt{6}(A - A_{\min} + (\widetilde{t}-t)\log(B))}{(\widetilde{t}-t)^{3/2}\sigma}\right] d\widetilde{t} = \frac{\mathrm{Erf}\left[\frac{(2r+\sigma^2)\sqrt{T-t}}{2\sigma\sqrt{2}}\right]}{B(2r+\sigma^2)} \tag{20}$$

with a suitable tolerance. Otherwise, it is possible to consider the whole unbounded A-domain $\equiv \mathbb{R}$, but with two infinite basis functions, the first and the last, as investigated in [4].

In order to numerically solve the Volterra integral Equation (15), we convert it into a discrete linear system of equations by means of collocation BEM. Hence we collocate (15) at points:

$$\overline{A}_i = \frac{A_i + A_{i-1}}{2}, \quad i = 1, \ldots, N_A \qquad \bar{t}_j = \frac{t_j + t_{j-1}}{2}, \quad j = 1, \ldots, N_t\,,$$

finally obtaining, for $i = 1, \ldots, N_A$, $j = 1, \ldots, N_t$:

$$\int_{\bar{t}_j}^T \int_{-\infty}^{+\infty} \frac{\sigma^2}{2} B^2 \sum_{k=1}^{N_t} \sum_{h=1}^{N_A} \alpha_h^{(k)} \psi_h(\widetilde{A}) \varphi_k(\widetilde{t}) G(B, \overline{A}_i, \bar{t}_j; B, \widetilde{A}, \widetilde{t})\, d\widetilde{A}\, d\widetilde{t} = -\int_{-\infty}^{+\infty} \int_0^B V(\widetilde{S}, \widetilde{A}, T) G(B, \overline{A}_i, \bar{t}_j; \widetilde{S}, \widetilde{A}, T)\, d\widetilde{S}\, d\widetilde{A}$$

i.e., in matrix notation,

$$\mathcal{A}\alpha = \mathcal{F}\,. \tag{21}$$

The unknown vector is:

$$\alpha = (\alpha^{(k)}|_{k=1,\ldots,N_t}) = \big((\alpha_h^{(k)}|_{h=1,\ldots,N_A})|_{k=1,\ldots,N_t}\big)\,,$$

and the matrix entries are equal to the case of fixed strike Asian options deeply investigated also from a computational point of view in [4,5]: for $i, h = 1, \ldots, N_A,\ j, k = 1, \ldots, N_t$, define $\xi = i - h,\ \xi = -N_A + 1, \ldots, N_A - 1$ and $\ell = k - j,\ \ell = 0, \ldots, N_t - 1$, thus obtaining:

$$\begin{aligned} \mathcal{A}_{ih}^{(jk)} &= \frac{\sigma B \Delta t}{4\sqrt{2\pi}} \int_{\frac{1}{2}-\frac{1}{2}H[\ell]}^{1} \frac{\exp\left\{-\left(\frac{2r+\sigma^2}{2\sqrt{2}\sigma}\right)^2 \Delta t(\tau+\ell-1/2)\right\}}{\sqrt{\Delta t(\tau+\ell-1/2)}} \left\{ \mathrm{Erf}\left[\frac{\sqrt{6}(\Delta A(\xi+\frac{1}{2}) + \Delta t(\tau+\ell-1/2)\log(B))}{\sigma \Delta t^{3/2}(\tau+\ell-1/2)^{3/2}}\right] \right. \\ &- \left. \mathrm{Erf}\left[\frac{\sqrt{6}(\Delta A(\xi-\frac{1}{2}) + \Delta t(\tau+\ell-1/2)\log(B))}{\sigma \Delta t^{3/2}(\tau+\ell-1/2)^{3/2}}\right] \right\} d\tau =: \mathcal{A}_{\xi}^{(\ell)} . \end{aligned} \tag{22}$$

As we are considering constant coefficients in (4), the fundamental solution (11) depends on $t, \widetilde{t}, A, \widetilde{A}$ only through the differences $t - \widetilde{t}$ and $A - \widetilde{A}$ implying that the matrix has a block-Toeplitz structure both in time and A-space:

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} \mathcal{A}^{(0)} & \mathcal{A}^{(1)} & \mathcal{A}^{(2)} & \cdots & \mathcal{A}^{(N_t-1)} \\ 0 & \mathcal{A}^{(0)} & \mathcal{A}^{(1)} & \cdots & \mathcal{A}^{(N_t-2)} \\ 0 & 0 & \mathcal{A}^{(0)} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathcal{A}^{(1)} \\ 0 & 0 & \cdots & 0 & \mathcal{A}^{(0)} \end{bmatrix} \\ \text{with} \quad \mathcal{A}^{(\ell)} &= \begin{bmatrix} \mathcal{A}_0^{(\ell)} & \mathcal{A}_{-1}^{(\ell)} & \mathcal{A}_{-2}^{(\ell)} & \cdots & \mathcal{A}_{-N_A+1}^{(\ell)} \\ \mathcal{A}_1^{(\ell)} & \mathcal{A}_0^{(\ell)} & \mathcal{A}_{-1}^{(\ell)} & \cdots & \mathcal{A}_{-N_A+2}^{(\ell)} \\ \mathcal{A}_2^{(\ell)} & \mathcal{A}_1^{(\ell)} & \mathcal{A}_0^{(\ell)} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathcal{A}_{-1}^{(\ell)} \\ \mathcal{A}_{N_A-1}^{(\ell)} & \mathcal{A}_{N_A-2}^{(\ell)} & \cdots & \mathcal{A}_1^{(\ell)} & \mathcal{A}_0^{(\ell)} \end{bmatrix} \quad \text{for } \ell = 0, \ldots, N_t - 1 \end{aligned} \tag{23}$$

so it is possible to adopt suitable strategies to save computational costs (as done in [13], even if this feature is not implemented in the code included here) and memory requirements.

A change in the payoff function is caught by the evaluation of the rhs entries: in the case of a floating strike call payoff, for $i = 1, \ldots, N_A$, $j = 1, \ldots, N_t$:

$$\mathcal{F}_i^{(j)} = -\int_{-\infty}^{+\infty} \int_0^B \max\left(\widetilde{S} - e^{\frac{\widetilde{A}}{T}}, 0\right) G(B, \overline{A}_i, \overline{t}_j; \widetilde{S}, \widetilde{A}, T)\, d\widetilde{S}\, d\widetilde{A} \tag{24}$$

$$\begin{aligned} &= \frac{B^{-\frac{2r-\sigma^2}{2\sigma^2}}}{2\sigma\sqrt{2\pi(T-\overline{t}_j)}} \int_0^B \widetilde{S}^{\frac{2r-3\sigma^2}{2\sigma^2}+\frac{T-\overline{t}_j}{2T}} \exp\left(-\frac{(T-\overline{t}_j)^2(2r+\sigma^2)^2 + 4\log^2\left(\frac{B}{\widetilde{S}}\right)}{8\sigma^2(T-\overline{t}_j)}\right) \\ &\left\{ -\widetilde{S}^{1-\frac{T-\overline{t}_j}{2T}} \mathrm{Erfc}\left[\sqrt{\frac{3}{2}} \frac{2\overline{A}_i + 2(T-\overline{t}_j)\log(B) - 2T\log(\widetilde{S}) - (T-\overline{t}_j)\log(\frac{B}{\widetilde{S}})}{\sigma(T-\overline{t}_j)^{\frac{3}{2}}}\right] \right. \\ &+ B^{\frac{T-\overline{t}_j}{2T}} \exp\left(\frac{\overline{A}_i}{T} + \frac{\sigma^2(T-\overline{t}_j)^3}{24\,T^2}\right) \\ &\left. \mathrm{Erfc}\left[\frac{12\,T\overline{A}_i + \sigma^2(T-\overline{t}_j)^3 - 6\,T\big(-2(T-\overline{t}_j)\log(B) + 2\,T\log(\widetilde{S}) + (T-\overline{t}_j)\log(\frac{B}{\widetilde{S}})\big)}{2\sqrt{6}\,T\sigma(T-\overline{t}_j)^{\frac{3}{2}}}\right] \right\} d\widetilde{S}, \end{aligned} \tag{25}$$

with Erfc the complementary of error function Erf.

The approximation of $\frac{\partial V}{\partial \widetilde{S}}(B, \widetilde{A}, \widetilde{t})$ through the resolution of (21) is an intermediate step for the final evaluation of the option price by (14):

$$
\begin{aligned}
V(S,A,t) \quad \approx \quad & -\frac{S^{-\frac{2r-\sigma^2}{2\sigma^2}}}{2\sigma\sqrt{2\pi(T-t)}} \int_0^B \widetilde{S}^{\frac{2r-3\sigma^2}{2\sigma^2}+\frac{T-t}{2T}} \exp\left(-\frac{(T-t)^2(2r+\sigma^2)^2 + 4\log^2\left(\frac{S}{\widetilde{S}}\right)}{8\sigma^2(T-t)}\right) \\
& \left\{-\widetilde{S}^{1-\frac{T-t}{2T}} \operatorname{Erfc}\left[\sqrt{\frac{3}{2}}\,\frac{2A+2(T-t)\log(S)-2T\log(\widetilde{S})-(T-t)\log(\frac{S}{\widetilde{S}})}{\sigma(T-t)^{\frac{3}{2}}}\right]\right. \\
+ \quad & S^{\frac{T-t}{2T}} \exp\left(\frac{A}{T} + \frac{\sigma^2(T-t)^3}{24\,T^2}\right) \\
& \left.\operatorname{Erfc}\left[\frac{12\,TA + \sigma^2(T-t)^3 - 6\,T\big(-2(T-t)\log(S) + 2\,T\log(\widetilde{S}) + (T-t)\log(\frac{S}{\widetilde{S}})\big)}{2\sqrt{6}\,T\sigma(T-t)^{\frac{3}{2}}}\right]\right\} d\widetilde{S} \qquad (26) \\
+ \quad & \frac{\sigma B}{4\sqrt{2\pi}} \left(\frac{B}{S}\right)^{\frac{2r-\sigma^2}{2\sigma^2}} \sum_{k=\mathrm{floor}[\frac{t}{\Delta t}]+1}^{N_t} \sum_{h=1}^{N_A} \alpha_h^{(k)} \int_{\max(t,t_{k-1})}^{t_k} \frac{1}{\sqrt{\widetilde{t}-t}} \exp\left(-\frac{(\widetilde{t}-t)^2(2r+\sigma^2)^2 + 4\log^2\left(\frac{S}{B}\right)}{8(\widetilde{t}-t)\sigma^2}\right) \\
& \left\{\operatorname{Erf}\left[\sqrt{\frac{3}{2}}\frac{2(A_h-A)-2(\widetilde{t}-t)\log(S)+(\widetilde{t}-t)\log\left(\frac{S}{B}\right)}{(\widetilde{t}-t)^{3/2}\sigma}\right]\right. \\
- \quad & \left.\operatorname{Erf}\left[\sqrt{\frac{3}{2}}\frac{2(A_{h-1}-A)-2(\widetilde{t}-t)\log(S)+(\widetilde{t}-t)\log\left(\frac{S}{B}\right)}{(\widetilde{t}-t)^{3/2}\sigma}\right]\right\} d\widetilde{t}\,.
\end{aligned}
$$

4.2. Finite Difference Methods

SABO is compared here (as in [5]) with two Finite Difference (FD) methods chosen among the wide class of FD methods available and deeply analyzed in [14]. This is because Equation (4) is proven to be hypoelliptic [10–12], a property that guarantees a smooth solution and that should benefit from approximations based on Taylor expansions.

The existence of the solution of Problem (4)–(5) and its Feynman–Kac representation (10) are proven involving stochastic arguments without the need for exact boundary conditions. Boundary conditions at $S = 0$ and for $A \to \pm\infty$ have to be empirically deduced analogously to what was done in [5,15]. At $S = B$, Condition (6) holds.

In order to apply FD methods, the first step is to shrink to a bounded domain $[0, B] \times [A_{\min}, A_{\max}]$ enclosing the option evaluation point (S^*, A^*) and then apply boundary conditions at the borders.

- We have chosen $A_{\min}$ less than or equal to the minimum between the value suggested in (20) and A^*. There is not a consistent condition valid at $A = -\infty$ in finance, and as a consequence, it is not easy to conceive of a proper condition at points $(S, A_{\min}, t)$. The use of the upwind method makes this condition unnecessary if $\log(S) > 0$ (as in the herein proposed examples). In fact, $\frac{\partial V}{\partial A}$ is approximated by a forward difference so that the boundary condition at $A_{\min}$ becomes useless; otherwise, if $\log(S) < 0$, a backward difference can be considered together with a condition on the derivative at $A = A_{\min}$, as for example $\frac{\partial^2 V}{\partial A^2}(S, A_{\min}, t) = 0$.
- The upper bound is set at:
$$A_{\max} = T\log(B)\,, \qquad (27)$$
the maximum value assumed by the stochastic process A_t defined in (1) if $S_t = B$ during the whole time interval $[0, T]$.

  The average $\exp(A/T)$ is a non-decreasing function of time; therefore, if $\exp(A/t) - B > 0$ at time t, then $\exp(A/T) - B > 0$ at maturity, and so, the option is worth nothing as shown by the Payoff Function (8). Hence, if $\log(S) > 0$, $\frac{\partial V}{\partial A}$ is approximated by a forward difference and the boundary condition at $A_{\max}$ is set:
$$V(S, A_{\max}, t) = 0\,. \qquad (28)$$

- At $S = 0$, Equation (4) is degenerate, but if the stochastic process $S_t = 0$ at any time, then the average asset price $\exp(A_t/t) = \textit{constant}$ and the option value V can be considered as independent of stochastic variables S and A, deducing from Equation (4) that:

$$\lim_{S\to 0}\left[\frac{\partial V}{\partial t}+\frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2}+rS\frac{\partial V}{\partial S}+\log(S)\frac{\partial V}{\partial A}-rV\right]=\lim_{S\to 0}\left[\frac{\partial V}{\partial t}-rV\right]=0\,.$$

  The ordinary differential equation:

$$\lim_{S\to 0}\left[\frac{\partial V}{\partial t}-rV\right]=0$$

  suggests the condition:

$$\lim_{S\to 0}V(S,A,t)=e^{-r(T-t)}\lim_{S\to 0}V(S,A,T)=0\,. \tag{29}$$

The second step, after the definition of the computational domain, is the definition of the grid. Using a (A,t)-grid as defined by (17) and (18) in $[A_{\min}, A_{\max}]\times[0,T]$, let us further introduce the S-grid in $[0,B]$:

$$\Delta S := \frac{B}{N_S},\quad N_S\in\mathbb{N}^+,\qquad S_i := i\Delta S,\quad i=0,\ldots,N_S \tag{30}$$

and define the approximated option value:

$$V_{i,h}^k :\approx V(S_i,A_h,t_k)\,.$$

- The values $V_{ih}^{N_t}$ can be found from the final condition:

$$V_{i,h}^{N_t}=V(S_i,A_h,T),\quad i=0,\cdots,N_S-1,\ h=0,\cdots,N_A\,;$$

- in the herein proposed examples, $\log(S_i)>0$, so everywhere, we apply a forward difference and, at the boundary $A=A_{\max}$, Condition (28):

$$V_{i,N_A}^k=0\qquad \text{for } k=0,\cdots,N_t-1,\ i=0,\cdots,N_S-1\,;$$

- at the boundary $S=0$, we apply Condition (29):

$$V_{0,h}^k=0\qquad h=0,\cdots,N_A-1,\ k=0,\cdots,N_t-1\,;$$

- at the boundary $S=B$, we apply the Boundary Condition (6), hence:

$$V_{N_S,h}^k=0,\quad h=0,\cdots,N_A,\ k=0,\cdots,N_t\,.$$

4.2.1. Finite Difference Method FD1

The first scheme FD1 is the very classical FD scheme where the derivatives of V in (4) are approximated by truncations of Taylor expansions:

- first order backward difference for the time derivative approximation:

$$\frac{\partial V}{\partial t}(S_i,A_h,t_k)=\frac{V_{i,h}^k-V_{i,h}^{k-1}}{\Delta t}+O(\Delta t)$$

- second order central difference for the S derivative approximation:

$$\frac{\partial V}{\partial S}(S_i, A_h, t_k) = \frac{V^k_{i+1,h} - V^k_{i-1,h}}{2\Delta S} + O(\Delta S^2)$$

- second order central difference for the S second order derivative approximation:

$$\frac{\partial^2 V}{\partial S^2}(S_i, A_h, t_k) = \frac{V^k_{i+1,h} - 2V^k_{i,h} + V^k_{i-1,h}}{\Delta S^2} + O(\Delta S^2)$$

- if $\log(S_i) > 0$, first order forward difference for the A derivative approximation:

$$\frac{\partial V}{\partial A}(S_i, A_h, t_k) = \frac{V^k_{i,h+1} - V^k_{i,h}}{\Delta A} + O(\Delta A)$$

We can now write down the discrete approximation of (4) at each point (S_i, A_h, t_k):

for $i = 1, \dots, N_S - 1$, $h = 0, \dots, N_A - 1$, $k = 0, \dots, N_t - 1$,

$$\tfrac{V^k_{i,h} - V^{k-1}_{i,h}}{\Delta t} + \tfrac{1}{2}\sigma^2 S_i^2 \tfrac{V^k_{i+1,h} - 2V^k_{i,h} + V^k_{i-1,h}}{\Delta S^2} + rS_i \tfrac{V^k_{i+1,h} - V^k_{i-1,h}}{2\Delta S} + \log(S_i) \tfrac{V^k_{i,h+1} - V^k_{i,h}}{\Delta A} - rV^k_{i,h} = 0 \tag{31}$$

and rearrange the scheme in a compact form:

$$V^{k-1}_{i,h} = a_i V^k_{i,h} + b_i V^k_{i+1,h} + c_i V^k_{i-1,h} + d_i V^k_{i,h+1}$$

where:

$$a_i = 1 - \Delta t\big(r + \sigma^2 i^2 + \tfrac{\log(S_i)}{\Delta A}\big)\,,\ b_i = \tfrac{\Delta t}{2}\big(ri + \sigma^2 i^2\big)\,, \qquad c_i = \tfrac{\Delta t}{2}\big(-ri + \sigma^2 i^2\big)\,, \qquad d_i = \tfrac{\Delta t}{\Delta A}\log(S_i)\,.$$

It is easy to see that the scheme, backward computing the new value $V^{k-1}_{i,h}$, is explicit in time. We want to remark about two more things: first, the weights depend only on i, therefore on the S variable, and second, the values $V^k_{i,0}$ having $A_{\min}$ as the coordinate give no contribution to the scheme.

4.2.2. Finite Difference Method FD2

The method FD2 is suggested in [16]. The PDE (4) is collocated at the points:

$$(S_i, A_{h+\frac{1}{2}}, t_{k+\frac{1}{2}}) := \Big(S_i, A_h + \frac{\Delta A}{2}, t_k + \frac{\Delta t}{2}\Big)\,.$$

Then, we use the following approximations, based on suitable Taylor expansions and standard finite difference approximations:

$$\begin{aligned}
\frac{\partial V}{\partial t}(S_i, A_{h+\frac{1}{2}}, t_{k+\frac{1}{2}}) &= \frac{1}{2}\Big(\frac{\partial V}{\partial t}(S_i, A_h, t_{k+\frac{1}{2}}) + \frac{\partial V}{\partial t}(S_i, A_{h+1}, t_{k+\frac{1}{2}})\Big) + O(\Delta A^2) \\
&= \frac{V^{k+1}_{i,h} - V^k_{i,h}}{2\Delta t} + \frac{V^{k+1}_{i,h+1} - V^k_{i,h+1}}{2\Delta t} + O(\Delta t^2 + \Delta A^2)\,,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial V}{\partial S}(S_i, A_{h+\frac{1}{2}}, t_{k+\frac{1}{2}}) &= \frac{1}{2}\Big(\frac{\partial V}{\partial S}(S_i, A_{h+1}, t_{k+1}) + \frac{\partial V}{\partial S}(S_i, A_h, t_k)\Big) + O(\Delta t^2 + \Delta A^2) \\
&= \frac{V^{k+1}_{i+1,h+1} - V^{k+1}_{i-1,h+1}}{4\Delta S} + \frac{V^k_{i+1,h} - V^k_{i-1,h}}{4\Delta S} + O(\Delta t^2 + \Delta A^2 + \Delta S^2)\,,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 V}{\partial S^2}(S_i, A_{h+\frac{1}{2}}, t_{k+\frac{1}{2}}) &= \frac{1}{2}\Big(\frac{\partial^2 V}{\partial S^2}(S_i, A_{h+1}, t_{k+1}) + \frac{\partial^2 V}{\partial S^2}(S_i, A_h, t_k)\Big) + O(\Delta t^2 + \Delta A^2) \\
&= \frac{V^{k+1}_{i-1,h+1} - 2V^{k+1}_{i,h+1} + V^{k+1}_{i+1,h+1}}{2\Delta S^2} + \frac{V^{k}_{i-1,h} - 2V^{k}_{i,h} + V^{k}_{i+1,h}}{2\Delta S^2} + O(\Delta t^2 + \Delta A^2 + \Delta S^2)\,,
\end{aligned}$$

$$V(S_i, A_{h+\frac{1}{2}}, t_{k+\frac{1}{2}}) = \frac{1}{2}\Big(V^k_{i,h} + V^{k+1}_{i,h+1}\Big) + O(\Delta t^2 + \Delta A^2)\,,$$

$$\begin{aligned}
\frac{\partial V}{\partial A}(S_i, A_{h+\frac{1}{2}}, t_{k+\frac{1}{2}}) &= \frac{1}{2}\Big(\frac{\partial V}{\partial A}(S_i, A_{h+\frac{1}{2}}, t_{k+1}) + \frac{\partial V}{\partial A}(S_i, A_{h+\frac{1}{2}}, t_k)\Big) + O(\Delta t^2) \\
&= \frac{V^{k+1}_{i,h+1} - V^{k+1}_{i,h}}{2\Delta A} + \frac{V^{k}_{i,h+1} - V^{k}_{i,h}}{2\Delta t} + O(\Delta t^2 + \Delta A^2)\,.
\end{aligned}$$

After substituting the above approximations into the PDE and discarding the error terms, we get the following equations for the approximate values of the option prices:

$$\overline{a}_i V^k_{i-1,h} + \overline{b}_i V^k_{i,h} + \overline{c}_i V^k_{i+1,h} = \overline{d}_i V^{k+1}_{i-1,h+1} + \overline{e}_i V^{k+1}_{i,h+1} + \overline{f}_i V^{k+1}_{i+1,h+1} + \overline{g}_i (V^k_{i,h+1} - V^{k+1}_{i,h}) \tag{32}$$

where, using the notation $\lambda = \frac{\Delta t}{\Delta S^2}$:

$$\begin{aligned}
&\overline{a}_i = \tfrac{\lambda}{2}\Big(- S_i^2\sigma^2 + rS_i\Delta S\Big)\,, \quad \overline{b}_i = \Big(1 + \lambda S_i^2\sigma^2 + \tfrac{\log(S_i)\Delta t}{\Delta A} + r\Delta t\Big)\,, \quad \overline{c}_i = -\tfrac{\lambda}{2}\Big(S_i^2\sigma^2 + rS_i\Delta S\Big)\,, \\
&\overline{d}_i = -a_i\,, \quad \overline{e}_i = \Big(1 - \lambda S_i^2\sigma^2 + \tfrac{\log(S_i)\Delta t}{\Delta A} - r\Delta t\Big)\,, \quad \overline{f}_i = -c_i\,, \quad \overline{g}_i = \Big(- 1 + \tfrac{\log(S_i)\Delta t}{\Delta A}\Big)\,.
\end{aligned}$$

The procedure for solving the option pricing equation is as follows:

1. Fill the values $V^{N_t}_{i,h}$, $i = 0, \cdots, N_S, h = 0, \cdots, N_A$ using the payoff function.
2. For each $k = N_t - 1 : -1 : 0$:
    (a) Apply the boundary condition at $A = A_{\max}$ to define V^k_{i,N_A}, for $i = 0, \cdots, N_S$.
    (b) For each $h = N_A - 1 : -1 : 0$, solve for the three-diagonal system in the unknowns the values $V^k_{i,h}$, for $i = 1, \cdots, N_S - 1$, using the boundary condition at $S = 0$ and $S = B$.

This is a time-explicit difference scheme, as well. If we need the option price only at $t = 0$, then it is not necessary to store the full matrix V of approximate option prices; in fact, we need only two levels, say V_{old} corresponding to $t = t_{k+1}$ and V_{new} corresponding to the current time level $t = t_k$. At the beginning, V_{old} is computed using the final condition, and at the end of each time step, the values of V_{new} are copied to V_{old}. Anyway, it requires the resolution of a linear system: the three-diagonal matrix $M = diag(\overline{a}, \overline{b}, \overline{c})$ can be assembled and factorized at the beginning, outside the cycles, depending only on the S-grid.

5. MATLAB® Code Implementing SABO

All the above-provided numerical results were obtained by codes developed with MATLAB® Release 2007b running on a laptop computer (CPU Intel i5, 4 Gb RAM). The code implementing the SABO algorithm is given below.

```
% Approximation of an up—and—out Asian call option with floating strike
% by SABO

close all
clear
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%data set
t0=0; %beginning of the contract
T=1; %expiry of the contract
r=0.035; %free risk interest rate
sigma=0.2; %volatility
B=150; %up barrier value
Sstar=[100,120,140]; %actual spot price
Astar=0; %actual geometric mean value
tstar=t0; %actual time instant of evaluation
prec=10^(-9); % precision required in Matlab "quad" function

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%discretization of time domain
Nt=20; %number of time steps
dt=(T-t0)/Nt; %time step length
t=[0:dt:T]; %time grid
tbar=(t(2:Nt+1)+t(1:Nt))/2; %time collocation points

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%discretization of A-domain
NA=Nt; %number of A-subintervals
AA=-(2*T^(3/2)*sigma)/sqrt(6)+Astar+T*log(B);
Amin=round(fzero(@(a) abs(...
    quad(@(tau) exp(-(tau*(2*r+sigma^2)^2)/(8*sigma^2))./...
    (2*sigma*sqrt(2*pi*tau)).*...
    erf(sqrt(6)*(Astar-a+tau*log(B))./(tau.^(3/2)*sigma)),...
    0,T-t0,10^-11)-...
    erf((2*r+sigma^2)*sqrt(T-t0)/(2*sqrt(2)*sigma))/(2*r+sigma^2))-prec,...
   AA)); %artificial lower bound of A-domain
%Amin=0;
AA=(2*T^(3/2)*sigma)/sqrt(6)+Astar+T*log(B);
Amax=round(fzero(@(a) abs(...
    quad(@(tau) exp(-(tau*(2*r+sigma^2)^2)/(8*sigma^2))./...
    (2*sigma*sqrt(2*pi*tau)).*...
    erf(sqrt(6)*(Astar-a+tau*log(B))./(tau.^(3/2)*sigma)),...
    0,T-t0,10^-11)+...
    erf((2*r+sigma^2)*sqrt(T-t0)/(2*sqrt(2)*sigma))/(2*r+sigma^2))-prec,...
   AA)); %artificial upper bound of A-domain
%Amax=5;
dA=(Amax-Amin)/NA; %A-subinterval length
A=[Amin:dA:Amax]; %A-grid
Abar=(A(2:NA+1)+A(1:NA))/2; %A collocation points
disp('truncated A-domain:')
Amin
Amax
tic
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%computation of matrix entries
M=zeros(NA,NA,Nt);
M_inf=zeros(NA,NA,Nt);
KK=3.5; %bound useful to erf
%------------- main diagonal --------------%
for i=1:NA
    for h=1:NA
        % search of integrand function zeros
        if ((sqrt(6)*(Abar(i)-A(h+1)+dt*log(B)*(10^-16)))/...
                (sigma*dt^(3/2)*(10^-16)^(3/2))-KK > 0) || ...
                ((sqrt(6)*(Abar(i)-A(h)+dt*log(B)*(1-0.5)))/...
                (sigma*dt^(3/2)*(1-0.5)^(3/2))+KK < 0)
            M(i,h,1)=0;
        else
            if (sqrt(6)*(Abar(i)-A(h+1)+dt*log(B)*(1-0.5)))/...
                    (sigma*dt^(3/2)*(1-0.5)^(3/2))-KK > 0
                Upper=fzero(@(s) ...
                    (sqrt(6)*(Abar(i)-A(h+1)+dt*log(B)*(s-0.5)))./...
                    (sigma*dt^(3/2)*(s-0.5).^(3/2))-KK,[0.5+10^-16,1]);
            else
                Upper=1;
            end
            if (sqrt(6)*(Abar(i)-A(h)+dt*log(B)*10^-16))/...
                    (sigma*dt^(3/2)*(10^-16)^(3/2))+KK < 0
                Lower=fzero(@(s)...
                    (sqrt(6)*(Abar(i)-A(h)+dt*log(B)*(s-0.5)))./...
                    (sigma*dt^(3/2)*(s-0.5).^(3/2))+KK,[0.5+10^-16,1]);
            else
                Lower=0.5+10^-16;
            end
            M(i,h,1)=sigma*B*dt/4/sqrt(2*pi)*quad(@(s) ...
                exp(-((2*r+sigma^2)/(2*sqrt(2)*sigma))^2*dt*(s-0.5))./...
                sqrt(dt*(s-0.5)).*...
                (erf((sqrt(6)*(Abar(i)-A(h)+dt*log(B)*(s-0.5)))./...
                (sigma*dt^(3/2)*(s-0.5).^(3/2)))-...
                erf((sqrt(6)*(Abar(i)-A(h+1)+dt*log(B)*(s-0.5)))./...
                (sigma*dt^(3/2)*(s-0.5).^(3/2)))),Lower,Upper,prec);
        end
    end
end
disp('diagonal block: built!')
%------------ secondary diagonals ------------%
for ell=1:Nt-1
    for i=1:NA
        for h=1:NA
            % search of integrand function zeros
            if ((sqrt(6)*(Abar(i)-A(h+1)+dt*log(B)*(ell-0.5)))/...
                    (sigma*dt^(3/2)*(ell-0.5)^(3/2))-KK > 0) || ...
                    ((sqrt(6)*(Abar(i)-A(h)+dt*log(B)*(1+ell-0.5)))/...
                    (sigma*dt^(3/2)*(1+ell-0.5)^(3/2))+KK < 0)
                M(i,h,ell+1)=0;
            else
                if (sqrt(6)*(Abar(i)-A(h+1)+dt*log(B)*(1+ell-0.5)))/...
                        (sigma*dt^(3/2)*(1+ell-0.5)^(3/2))-KK > 0
                    Upper=fzero(@(s) ...
                      (sqrt(6)*(Abar(i)-A(h+1)+dt*log(B)*(s+ell-0.5)))./...
                      (sigma*dt^(3/2)*(s+ell-0.5).^(3/2))-KK,[0,1]);
                else
                    Upper=1;
                end
                if (sqrt(6)*(Abar(i)-A(h)+dt*log(B)*(ell-0.5)))/...
                        (sigma*dt^(3/2)*(ell-0.5)^(3/2))+KK < 0
                    Lower=fzero(@(s)...
                        (sqrt(6)*(Abar(i)-A(h)+dt*log(B)*(s+ell-0.5)))./...
                        (sigma*dt^(3/2)*(s+ell-0.5).^(3/2))+KK,[0,1]);
                else
                    Lower=0;
                end
                M(i,h,ell+1)=sigma*B*dt/4/sqrt(2*pi)*quad(@(s) ...
                   exp(-((2*r+sigma^2)/(2*sqrt(2)*sigma))^2*dt*...
                   (s+ell-0.5))./sqrt(dt*(s+ell-0.5)).*...
                   (erf((sqrt(6)*(dA*(i-h+0.5)+dt*log(B)*(s+ell-0.5)))./...
                   (sigma*dt^(3/2)*(s+ell-0.5).^(3/2)))-...
                   erf((sqrt(6)*(dA*(i-h-0.5)+dt*log(B)*(s+ell-0.5)))./...
                   (sigma*dt^(3/2)*(s+ell-0.5).^(3/2)))),...
                   Lower,Upper,prec);
            end
        end
    end
```

```
end
disp('matrix: built!')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%computation of rhs entries
Rhs=zeros(NA,1,Nt);
%-----------------------------------------%
for j=1:Nt
    for i=1:NA
        Rhs(i,1,j)=1/(2*sigma*sqrt(2*pi))*...
            quad(@(S) ...
            exp(-((T-tbar(j))^2*(2*r+sigma^2)^2+4*log(B./S).^2+...
            4*sigma^2*(T-tbar(j))*log(T-tbar(j)))/...
            (8*sigma^2*(T-tbar(j)))+...
            ((2*r-3*sigma^2)/(2*sigma^2)+(T-tbar(j))/(2*T))*log(S)-...
            (2*r-sigma^2)/(2*sigma^2)*log(B)).*...
            (-S.^((T+tbar(j))/(2*T)).*...
            erfc(sqrt(1.5)*(2*Abar(i)+2*(T-tbar(j))*log(B)-2*T*log(S)-...
            (T-tbar(j))*log(B./S))/(sigma*(T-tbar(j))^(3/2)))+...
            B^((T-tbar(j))/(2*T))*exp(Abar(i)/T+(sigma^2*(T-tbar(j))^3)/...
            (24*T^2))*(erfc((12*T*Abar(i)+sigma^2*(T-tbar(j))^3-6*T*...
            (-2*(T-tbar(j))*log(B)+2*T*log(S)+(T-tbar(j))*log(B./S)))/...
            (2*sqrt(6)*T*sigma*(T-tbar(j))^(3/2))))),0,B,prec);
    end
end
disp('rhs: built!')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%linear system resolution by backward substitution
alpha(:,1,Nt)=M(:,:,1)\Rhs(:,1,Nt);
for i=Nt-1:-1:1
    alpha(:,1,i)=Rhs(:,1,i);
    for j=i+1:Nt
        alpha(:,1,i)=alpha(:,1,i)-M(:,:,j-i+1)*alpha(:,1,j);
    end
    alpha(:,1,i)=M(:,:,1)\alpha(:,1,i);
end
disp('BIE solution: built!')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%computation of solution in domain nodes (Sstar,Astar) at time tstar
V=zeros(length(Sstar),length(Astar));
V1=zeros(length(Sstar),length(Astar));
F=zeros(NA,Nt);
for jS=1:length(Sstar)
    if Sstar(jS)>=B
        V(jS,:)=zeros(1,length(Astar));
    else
        for jA=1:length(Astar)
        %------------- first integral -------------%
            V1(jS,jA)=-1/(2*sigma*sqrt(2*pi))*...
            quad(@(S) ...
            exp(-((T-tstar)^2*(2*r+sigma^2)^2+4*log(Sstar(jS)./S).^2+...
            4*sigma^2*(T-tstar)*log(T-tstar))/(8*sigma^2*(T-tstar))+...
            ((2*r-3*sigma^2)/(2*sigma^2)+(T-tstar)/(2*T))*log(S)-...
            (2*r-sigma^2)/(2*sigma^2)*log(Sstar(jS))).*...
            (-S.^((T+tstar)/(2*T)).*erfc(sqrt(1.5)*(2*Astar(jA)+...
            2*(T-tstar)*log(Sstar(jS))-2*T*log(S)-...
            (T-tstar)*log(Sstar(jS)./S))/(sigma*(T-tstar)^(3/2)))+...
            Sstar(jS)^((T-tstar)/(2*T))*exp(Astar(jA)/T+...
            (sigma^2*(T-tstar)^3)/(24*T^2))*erfc((12*T*Astar(jA)+...
            sigma^2*(T-tstar)^3-6*T*(-2*(T-tstar)*log(Sstar(jS))+...
            2*T*log(S)+(T-tstar)*log(Sstar(jS)./S)))/...
            (2*sqrt(6)*T*sigma*(T-tstar)^(3/2)))),0,B,prec);
            disp('post-pro: first integral!')
            %——————— sum ———————%
            for kt=floor(tstar/dt)+1:Nt
                Lower=max(tstar,t(kt));
                Upper=t(kt+1);
                for hA=1:NA
                    F(hA,kt)=(sigma*B)/(4*sqrt(2*pi))*...
                        quad(@(s) exp(-((s-tstar).^2*(2*r+sigma^2)^2+...
                        4*log(Sstar(jS)/B)^2+4*sigma^2*(s-tstar).*...
                        log(s-tstar))./(8*(s-tstar)*sigma^2)+...
                        ((2*r-sigma^2)/(2*sigma^2))*log(B/Sstar(jS))).*...
                        (erf(sqrt(1.5)*(2*(A(hA+1)-Astar(jA))-...
                        2*(s-tstar)*log(Sstar(jS))+(s-tstar)*...
                        log(Sstar(jS)/B))./((s-tstar).^(1.5)*sigma))-...
                        erf(sqrt(1.5)*(2*(A(hA)-Astar(jA))-2*(s-tstar)*...
                        log(Sstar(jS))+(s-tstar)*log(Sstar(jS)/B))./...
                        ((s-tstar).^(1.5)*sigma))),Lower,Upper,prec);
                    V(jS,jA)=V(jS,jA)+alpha(hA,1,kt)*F(hA,kt);
                end
            end
            V(jS,jA)=V(jS,jA)+V1(jS,jA);
        end
    end
end
toc

dataset({[Sstar',V] 'S','V'})
%plot(Sstar',V) %graph of option values as function of asset values
```

Author Contributions: The Authors equally contributed to this work.

Acknowledgments: The dissemination of this research was supported by the INdAM-GNCS group (Italian National Group of Scientific Computing).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hull, J. *Options, Futures, and Other Derivatives*, 8th ed.; Pearson-Prentice Hall: Upper Saddle River, NJ, USA, 2011.
2. Boyle, P.; Potapchik, A. Prices and sensitivities of Asian options: A survey. *Insur. Math. Econ.* **2008**, *42*, 189–211. [CrossRef]
3. Cao, G.; Wang, Y. Risk-neutral pricing for geometric average Asian options with floating strike. *J. Chin. Acad. Sci.* **2015**, *32*, 13–17.
4. Aimi, A.; Guardasoni, C. Collocation Boundary Element Method for the pricing of Geometric Asian Options. *Eng. Anal. Bound. Elem.* **2017**, *92*, 90–100. [CrossRef]
5. Aimi, A.; Diazzi, L.; Guardasoni, C. Numerical Pricing of Geometric Asian Options with Barriers. *Math. Methods Appl. Sci.* **2018**, submitted.
6. Guardasoni, C.; Sanfelici, S. A Boundary Element approach to barrier option pricing in Black-Scholes framework. *Int. J. Comput. Math.* **2016**, *93*, 696–722. [CrossRef]
7. Guardasoni, C. Semi-Analytical method for the pricing of Barrier Options in case of time-dependent parameters (with Matlab codes). *Commun. Appl. Ind. Math.* **2018**, *9*, 42–67. [CrossRef]
8. Ballestra, L.; Pacelli, G. A boundary element method to price time-dependent double barrier options. *Appl. Math. Comput.* **2011**, *218*, 4192–4210. [CrossRef]
9. Wilmott, P. *On Quantitative Finance*; John Wiley and Sons: Hoboken, NJ, USA, 2000.
10. Barucci, E.; Polidoro, S.; Vespri, V. Some results on partial differential equations and Asian options. *Math. Model. Methods Appl. Sci.* **2001**, *11*, 475–497. [CrossRef]
11. Polidoro, S. Uniqueness and representation theorems for solutions of Kolmogorov-Fokker-Planck equations. *Rendiconti di Matematica e delle sue Applicazioni* **1995**, *15*, 535–560.
12. Hörmander, L. Hypoelliptic second order differential equations. *Acta Math.* **1967**, *119*, 147–171. [CrossRef]

13. Aimi, A.; Diligenti, M.; Guardasoni, C. Energetic BEM for the numerical solution of 2D hard scattering problems of damped waves by open arcs. *Springer INdAM Series* **2018**, accepted.
14. Hugger, J. A fixed strike Asian option and comments on its numerical solution. *ANZIAM J.* **2004**, *45*, C215–C231. [CrossRef]
15. Hugger, J. Wellposedness of the boundary value formulation of a fixed strike Asian option. *J. Comput. Appl. Math.* **2006**, *185*, 460–481. [CrossRef]
16. Kangro, R. *Lecture Notes on Computational Finance*; Technical Report; University of Tartu: Tartu, Estonia, 2011.