Efficient pricing of Asian options by the PDE approach.

François Dubois[1], Tony Lelièvre[2]
[1] Conservatoire National des Arts et Métiers, 292 rue Saint-Martin, 75003, Paris, France.
[2] CERMICS, Ecole Nationale des Ponts et Chaussées, 6 & 8 Av. Pascal, 77455 Champs-sur-Marne, France.
E-mail: fdubois@cnam.fr, lelievre@cermics.enpc.fr

October 2004

Abstract

We consider the partial differential equation proposed by Rogers and Shi in [8] and explain the main difficulties encountered when applying standard numerical schemes on this PDE as such. We then propose a scheme which is able to produce very quickly (in less than one second on a PC equipped with a 1 GHz Intel Pentium III microprocessor) accurate results (at least 5 digits of precision). We compare our approach with the schemes proposed in the literature.

Keywords: Asian options, Partial Differential Equation, characteristic method.

1 Introduction

In this article, we would like to explain why standard techniques do not yield accurate results for the solution of the partial differential equation ruling the price of an Asian option (see [8]):

$$\left\{ \begin{array}{l} \dfrac{\partial f}{\partial t} + \dfrac{\sigma^2 \xi^2}{2} \dfrac{\partial^2 f}{\partial \xi^2} - \left(\dfrac{1}{T} + r\xi \right) \dfrac{\partial f}{\partial \xi} = 0, \\ f(T, \xi) = \phi(\xi). \end{array} \right. \tag{1}$$

where $\xi \geq 0$ and the expression of ϕ depends on the payoff of the option (see Section 2 for details). We will focus on numerical methods based on this PDE. Let us also mention some Monte Carlo approaches (see [5]), some methods based on analytical or semi-analytical solutions (see [11, 3]) and also some approximation methods (see [7, 9]). The advantage of the PDE approach is that it is generally faster than Monte Carlo methods, and that it gives the results for all initial prices S_0 (and even for all strikes K or all maturities T in some cases). The drawback is usually that the numerical methods are more complicated to implement for PDE, but we give here a numerical scheme which is really simple to code. This scheme has been implemented in the software PREMIA (see routine *fd_fixedasian_rodgershi2.c* in [1]). The method used in this paper is based on the PDE (8) already derived by Vecer [10]. We will derive this PDE using a different method and give some generalizations. We will also show how the numerical approach used in [10] can be made more efficient.

2 The model

We adopt the standard Black and Scholes model (see [4]) with a risky asset whose price at time t is S_t and a no-risk asset whose price at time t is S_t^0, such that :

$$dS_t = S_t(\mu \, dt + \sigma dB_t), \quad dS_t^0 = rS_t^0 \, dt.$$

The process B_t is a standard Brownian motion defined on a probability space $(\Omega, \mathcal{F}, \mathcal{F}_t, \mathbb{Q})$, and μ, r (the interest rate), $\sigma > 0$ (the volatility) are three constants. We introduce the stochastic process $W_t = B_t + \frac{\mu - r}{\sigma} t$. Under the neutral risk probability $\mathbb{P}$, we know that S_t/S_t^0 is a martingale and that W_t is a Brownian motion. The process S_t is solution of the following stochastic differential equation under $\mathbb{P}$:

$$dS_t = S_t(r\,dt + \sigma dW_t). \tag{2}$$

We are interesting in computing the price of an Asian option with maturity T, which means that the option payoff $g(S, A)$ depends on the price S_T of the risky asset and on the mean A_T of the price S_t defined by $A_t = \frac{1}{t}\int_0^t S_u\,du$. It is well-known that in this model, the price of the option at time t is given by:

$$e^{-r(T-t)}\mathbb{E}(g(S_T, A_T)|\mathcal{F}_t).$$

Moreover, one can check that if V is solution of the following PDE (see [12]):

$$\begin{cases} \dfrac{\partial V}{\partial t} + \dfrac{\sigma^2 S^2}{2}\dfrac{\partial^2 V}{\partial S^2} + rS\dfrac{\partial V}{\partial S} + \dfrac{1}{t}(S - A)\dfrac{\partial V}{\partial A} - rV = 0, \\ V(T, S, A) = g(S, A). \end{cases} \tag{3}$$

then $e^{-rt}V(t, S_t, A_t)$ is a $\mathbb{P}$-martingale so that the price at time t is given by:

$$e^{-r(T-t)}\mathbb{E}(g(S_T, A_T)|\mathcal{F}_t) = V(t, S_t, A_t).$$

The PDE (3) is difficult to solve as such since the parabolic operator is degenerate in the A-variable (see [12]).

However, as remarked by Rogers and Shi in [8], for fixed strike call ($g(S, A) = (A-K)^+$) or fixed strike put ($g(S, A) = (K-A)^+$), it is possible to reduce (3) to the PDE (1). Indeed, if f is solution of (1) with $\phi(\xi) = \max(-\xi, 0) = \xi^-$ (resp. $\phi(\xi) = \max(\xi, 0) = \xi^+$), then

$$V(t, S, A) = Sf\left(t, \frac{K - tA/T}{S}\right) \tag{4}$$

is solution of (3) with $g(S, A) = (A-K)^+$ (resp. $g(S, A) = (K-A)^+$). This reduction of (3) to (1) is also possible for floating strike call ($g(S, A) = (S - A)^+$) (resp. for floating strike put ($g(S, A) = (A - S)^+$)) by setting $V(t, S, A) = Sf\left(t, -\frac{tA}{TS}\right)$ and $\phi(\xi) = (1 + \xi)^-$ (resp. $\phi(\xi) = (1 + \xi)^+$). For a derivation of an equation similar to (1) using an interpretation of Asian options as options on a traded account, see [10].

Notice that PDE (3) is more general since the reduction to (1) is only possible for specific payoffs. Moreover, one can also notice that solving (3) for different maturities T does not imply the need to recompute the whole solution, contrary to (1). In the following, we focus on computing numerical solutions to (1) for a call:

$$\phi(\xi) = \xi^-. \tag{5}$$

Notice that the solution with $\phi(\xi) = \xi^+$ (put) can be then obtained by the call-put parity:

$$\text{Price of the put}(t = 0, S_0) = \text{Price of the call}(t = 0, S_0) - e^{-rT}\left(\frac{S_0}{rT}\left(e^{rT} - 1\right) - K\right).$$

3 The numerical scheme

The question we want to address is the following. When one wants to compute the solution of the classical Black-Scholes equation:

$$\begin{cases} \dfrac{\partial P}{\partial t} + \dfrac{\sigma^2 S^2}{2}\dfrac{\partial^2 P}{\partial S^2} + rS\dfrac{\partial P}{\partial S} - rP = 0, \\ P(T,S) = \psi(S), \end{cases} \tag{6}$$

a simple finite difference scheme leads to very satisfactory result.

On the other hand, when one uses a simple finite difference scheme on (1), one obtains very bad results, especially when the volatility σ is small (see Table 1). These bad results are due to the fact that when ξ is close to zero, the advective term $\left(\frac{1}{T} + r\xi\right)$ is larger than the diffusion term $\frac{\sigma^2\xi^2}{2}$.

parameters	reference value	centered scheme
$\sigma = 0.3$, $I = 100$, $N = 1000$	7.3076	7.295
$\sigma = 0.3$, $I = 600$, $N = 1000$	7.3076	7.3043
$\sigma = 0.3$, $I = 600$, $N = 500$	7.3076	7.3041
$\sigma = 0.3$, $I = 1000$, $N = 200$	7.3076	7.3059
$\sigma = 0.05$, $I = 600$, $N = 500$	1.6970	1.783

Table 1: Results obtained with a finite element scheme, or, equivalently, a centered scheme for the advection operator. The numbers I and N denote respectively the number of space steps and the number of time steps. Values of other parameters: $S_0 = 100$, $K = 100$, $r = 0.02$, $T = 1$. Table from [6].

What we propose to solve this problem is to use a characteristic method (based on the solution of $\frac{d\xi}{dt} = -1/T$), to get rid of the term $1/T$. This means that we perform the change of variable:

$$g(t,x) = f(t, x - t/T). \tag{7}$$

One can easily show that g is solution of:

$$\begin{cases} \dfrac{\partial g}{\partial t} + \dfrac{\sigma^2 (x - t/T)^2}{2}\dfrac{\partial^2 g}{\partial x^2} - r\,(x - t/T)\,\dfrac{\partial g}{\partial x} = 0, \\ g(T,x) = \phi(x-1) = (1-x)^+. \end{cases} \tag{8}$$

Notice that the same equation has been obtained by Vecer in [10] by using some financial arguments. Our approach is different. For example, it can be generalized in order to completely get rid of the advective term by solving $\frac{d\xi}{dt} = -r\xi - 1/T$ and therefore by considering (in the case $r \neq 0$):

$$h(t,y) = f\left(t, \frac{1}{rT}\left(y e^{r(T-t)} - 1\right)\right). \tag{9}$$

One can easily show that h is then solution of:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\sigma^2}{2}\left(y - e^{-r(T-t)}\right)^2 \dfrac{\partial^2 h}{\partial y^2} = 0, \\ h(T,y) = \frac{1}{rT}(1-y)^+. \end{cases} \tag{10}$$

Following the same method as explained below, one can then obtain even more accurate results on a geometric space-time mesh (we refer to [2] for details and some numerical results). We have here restricted ourselves to the simple change of variable (7) since it leads to an easier scheme to implement. Notice that this can also be extended to the case when the interest rate is a known function of time $r(t)$ by solving $\frac{d\xi}{dt} = -r(t)\xi - 1/T$ either analytically (as in this paper) or numerically.

The PDE (8) satisfied by g is such that when the advective term $r\,(x - t/T)$ is small, the diffusion term $\dfrac{\sigma^2(x-t/T)^2}{2}$ is also small. The drawback of this change of variable is that the advective and diffusion terms now depend on time: this means that, once the problem is discretized, the matrices are computed and inverted at each time step.

The fact that (see [8]) $\forall \xi \leq 0$,

$$f(t,\xi) = \frac{1}{rT}(1 - e^{-r(T-t)}) - \xi e^{-r(T-t)} \tag{11}$$

yields for g that $\forall x \leq t/T$,

$$g(t,x) = \frac{1}{rT}(1 - e^{-r(T-t)}) - (x - t/T)e^{-r(T-t)}. \tag{12}$$

To understand formula (11) from a PDE point of view, one can notice that f given by (11) is solution to (1) with $\phi(\xi) = -\xi$, and that, due to the fact that the diffusion term is null for $\xi = 0$ and that the advective term is negative, the solution to (1) for $\phi(\xi) = \xi^-$ on $\xi \leq 0$ is the same as the solution to (1) for $\phi(\xi) = -\xi$ on $\xi \leq 0$.

To discretize (8), we use a Crank-Nicolson time scheme, with a uniform time step $\delta t = T/N$ (N denotes the number of timesteps). We want to use the fact that g is analytically known on $x \leq t/T$ (see Formula (12)). Therefore, in order that the mesh properly discretizes the border $x = t/T$, we also use N space steps to discretize the space interval $(0,1)$ (see Figure 1). The space step is therefore $\delta x = 1/N$. We then complete the mesh be adding J intervals on the right hand side of $x = 1$, so that $x \in (0, x_{max})$ with $x_{max} = (N + J)\,\delta x$.

Notice that at time $t_n = n\delta t$, the number of unknowns is $(N + J - n)$. This means that the sizes of the matrices we build and invert depend on the timestep. Since the mesh

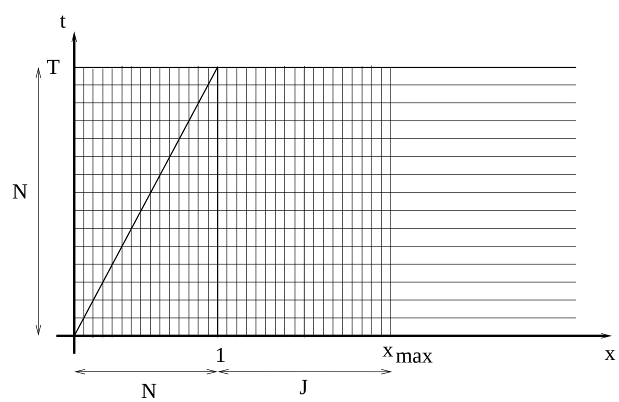

Figure 1: The function g and its computational domain.

is uniform, it is really easy to discretize the operators involving derivatives of x. What we use is the following equation on g, equivalent to (8):

$$\begin{cases} \dfrac{\partial g}{\partial t} + \dfrac{\sigma^2}{2}\dfrac{\partial}{\partial x}\left((x - t/T)^2 \dfrac{\partial g}{\partial x}\right) - (r + \sigma^2)\,(x - t/T)\,\dfrac{\partial g}{\partial x} = 0, \\ g(T,x) = \phi(x-1) = (1-x)^+. \end{cases} \tag{13}$$

so that one can use the following approximations:

$$\frac{\partial}{\partial x}\left((x-t/T)^2\frac{\partial g}{\partial x}\right)(t,x_i)\simeq$$
$$\frac{1}{\delta x}\left((x_{i+1/2}-t/T)^2\frac{g(t,x_{i+1})-g(t,x_i)}{\delta x}-(x_{i-1/2}-t/T)^2\frac{g(t,x_i)-g(t,x_{i-1})}{\delta x}\right),$$

$$(x-t/T)\frac{\partial g}{\partial x}(t,x_i)\simeq$$
$$\frac{1}{2}\left((x_{i+1/2}-t/T)\frac{g(t,x_{i+1})-g(t,x_i)}{\delta x}+(x_{i-1/2}-t/T)\frac{g(t,x_i)-g(t,x_{i-1})}{\delta x}\right),$$

with $x_i = i\,\delta x$ and $x_{i+1/2} = (i+1/2)\,\delta x$. The matrices obtained are tridiagonal, so that they can be inverted with LU method with linear complexity.

As far as boundary conditions are concerned, we use a Dirichlet boundary condition on $x = t/T$ using (12) and a "artificial" zero Neumann boundary condition on $x = x_{max}$.

Finally, we use interpolation of degree 3 in space to compute the price, and of degree 2 in space to compute the delta. These quantities are given by (see formula (4)):

$$\text{price} = S_0 f\left(0,\frac{K}{S_0}\right) = S_0 g\left(0,\frac{K}{S_0}\right), \tag{14}$$

$$\text{delta} = f\left(0,\frac{K}{S_0}\right) - \frac{K}{S_0}\frac{\partial f}{\partial x}\left(0,\frac{K}{S_0}\right) = g\left(0,\frac{K}{S_0}\right) - \frac{K}{S_0}\frac{\partial g}{\partial x}\left(0,\frac{K}{S_0}\right). \tag{15}$$

We would like to emphasize that our numerical scheme:

- can handle negative or zero interest rate r, which is important if one takes into account some dividend rate d (since in this case, r is replaced by $r-d$ and formulas (14) and (15) for price and delta are multiplied by e^{-dT}),
- can also handle the case of time-dependent interest rate r and volatility σ.

Let us precise this last point. If one considers a time-dependent interest rate $r(t)$ and volatility $\sigma(t)$, it is easy to show that a solution to (2) with $r(t)$ and $\sigma(t)$ can be obtained by considering a solution to (1) with $r(t)$ and $\sigma(t)$. Moreover, in the case of a call, for example, ($\phi(\xi) = \xi^-$), we have also an analytical solution for $\xi \leq 0$ given by $f(t,\xi) = \frac{1}{T}\int_t^T \exp(-\int_s^T r_u\,du)\,ds - \xi\exp(-\int_t^T r_s\,ds)$. This shows that the numerical method described above can be easily adapted to the case of time-dependent interest rate r and volatility σ.

Notice that our method cannot apply to (t,S,A)-dependent interest rate and volatility, except in the case of $\sigma(t,S,A) = \eta\left(t,\frac{K-tA/T}{S}\right)$ and $r(t,S,A) = \rho\left(t,\frac{K-tA/T}{S}\right)$ (for fixed strike call and put). It seems that to handle a general (t,S,A)-dependency, one needs to solve directly (3).

4 Some numerical results

All the presented numerical results and the computational times have been obtained with a program written in C and run on a PC equipped with a 1 GHz Intel Pentium III microprocessor.

We express J (the number of intervals on $x \geq 1$) as a function of N, such that the results do not change with larger J. We have found (see Table 2) that the value $J = N/2$ is sufficient to guaranty the fact that the price does not depend on the position of x_{max}.

	$J = 50$	$J = 100$	$J = 500$	$J = 1000$
default values	1.697260346	1.696871955	1.696871955	1.696871955
$T = 0.25$	0.706693703	0.7066937038	0.7066937038	0.7066937038
$T = 0.25$, $N = 100$	0.6668349075	0.6668349075	0.6668349075	0.6668349075
$K = 104$	0.2812271443	0.24000941	0.2400092691	0.2400092691

Table 2: Comparison of the results for different values of J. Default values of parameters: $T = 1$, $\sigma = 0.05$, $r = 0.02$, $S_0 = 100$, $K = 100$, $N = 1000$.

We have also performed a rate of convergence analysis (see Table 3). We have found that both the price and the delta values converge with a rate $O\left(\frac{1}{N^2}\right)$. The fact that the delta converges with the same rate as the price is quite surprising since the delta value (see formula (15)) contains derivative of g. Notice that in Table 3, a precision of 4 digits both for price and delta can be obtained with $N = 800$, which corresponds to a computational time of 0.4 s.

absolute error on	$N = 100$	$N = 200$	$N = 400$	$N = 800$
price	.019144	.004697	.001168	.000291
delta	.0018231	.0003120	.0000603	.0000128

Table 3: Rate of convergence. Values of parameters: $T = 1.$, $\sigma = 0.05$, $r = 0.02$, $S_0 = 100$, $K = 100$, $J = N/2$. We give the absolute error between the price and delta obtained and the reference values ($N = 20000$): price= 1.697058, delta= 0.6334899.

Finally, we give in Table 4 a few comparisons of the results obtained with our method and other methods. The prices displayed for our method are converged (the digits given in Table 4 do not change for larger N). We also indicate the number of time steps $N \geq 300$ needed to obtain at least the 5 digits of accuracy given. The computational times are given in Table 5. Notice that, as expected, the computational time is $O(N^2)$ (since for N timesteps, we use $3N/2$ space steps and the matrices can be inverted with linear complexity).

The results from Zvan et al. come from [12] (see Table 2 in [12]): the indicated computational time indicated in [12] is about 20 s on a DEC Alpha. The results from Večeř come from [10] and are obtained in a few seconds. The lower and upper bounds are obtained with the Premia software following the method of Thompson (see [1, 9]): the computational time is about 5 s on a PC equipped with a 1 GHz Intel Pentium III microprocessor.

One can see from these comparisons that our method is accurate both for small and large volatilities. For any values of the parameters we have considered, we have obtained at least 5 digits of precision in less than one second. It seems also that our method is faster than the other, but this would have to be checked carefully since it depends on the computer used. We can affirm that it is at least faster than the method of Thompson (see [9]) based on approximations, since we have tested both methods on the same computer.

We can also conclude from these experiments that, as expected, the accuracy for a given N is better for strikes less than S_0, as well as for large volatilities.

σ	K	Our method	Zvan et al.	Večeř	Thompson (low)	Thompson (up)
0.05	95	11.09409 ($N = 300$)	11.094	11.094	11.094094	11.094096
	100	6.7943 ($N = 1000$)	6.793	6.795	6.794354	6.794465
	105	2.7444 ($N = 3000$)	2.748	2.744	2.744406	2.744581
0.30	90	16.512 ($N = 300$)	16.514	16.516	16.512024	16.523720
	100	10.209 ($N = 300$)	10.210	10.215	10.208724	10.214085
	110	5.7304 ($N = 1000$)	5.729	5.736	5.728161	5.735488

Table 4: Comparisons of the prices obtained with other methods. Values of parameters: $T = 1$, $r = 0.15$, $S_0 = 100$, $J = N/2$. For our method, we give the number of timesteps $N \geq 300$ needed to obtain at least 5 digits of precision.

N	300	1000	3000	6000
computational time	0.05 s	0.56 s	5.2 s	27 s

Table 5: Computational times for our method for some N and $J = N/2$ (computations made on a PC equipped with a 1 GHz Intel Pentium III).

5 Conclusion

In conclusion, we have derived a very accurate and fast numerical method to solve the Rodger-Shi PDE. We obtain, in less than one second (on a PC equipped with a 1 GHz Intel Pentium III microprocessor), at least 5 digits of accuracy both for large or small volatilities ($\sigma = 0.05$ or $\sigma = 0.30$) and for in-the-money or out-of-the-money computations. The results are also very accurate for the delta. Our method is easy to implement and can handle time-dependent interest rates or volatilities. It also works with null or negative interest rates, which is of interest if one takes into account some dividend rates.

Acknowledgments: The authors would like to thank Bernard Lapeyre for stimulating and enlightening discussions and also Oualid Benallou for his work as a student on this subject.

References

[1] Premia: an option pricer. http://cermics.enpc.fr/~premia/index.html.

[2] F. Dubois. Différences finies caractéristiques pour les options asiatiques, 2003. http://www.asci.fr/dubois/jan03/Options.pdf.

[3] G. Fusai. Pricing Asian options via Fourier and Laplace transforms. *Journal of Computational Finance*, 7(3), 2004.

[4] D. Lamberton and B. Lapeyre. *Introduction au calcul stochastique appliqué à la finance.* Ellipses, 1997.

[5] B. Lapeyre and E. Temam. Competitive Monte Carlo methods for the pricing of Asian options. *Journal of Computational Finance*, 5(1), 2001.

[6] T. Lelièvre. Pricing des options asiatiques, 2000. http://cermics.enpc.fr/~lelievre/rapports/RapportAsia.ps.

[7] E. Lévy. Pricing European average rate currency options. *J. of International Money and Finance*, 11:474–491, 1992.

[8] L.C.G. Rogers and Z. Shi. The value of an Asian option. *J. Appl. Probability*, 32:1077–1088, 1995.

[9] G.W.P. Thompson. Fast narrow bounds on the value of Asian options. *Working paper Judge Institute U. of Cambridge*, 1999.

[10] J. Večeř. A new PDE approach for pricing arithmetic average Asian options. *Journal of Computational Finance*, 4(4):105–113, 2001.

[11] J.E. Zhang. A semi-analytical method for pricing and hedging continuously-sampled arithmetic avarage rate options. *Journal of Computational Finance*, 5(1):59–79, 2001.

[12] R. Zvan, P.A. Forsyth, and K. Vetzal. Robust numerical methods for PDE models of Asian options. *Journal of Computational Finance*, 1(2):39–78, 1998.