Pricing of an American option

Module :
FQ301 (PDE in finance)

Realized by :
Samer SAID
Kais BEN ISMAIL

Encadrant :
O. BOKANOWSKI

December 11, 2018

Contents

Introduction

We consider the value of the American option : $V(t,x) = sup_{\tau \in \mathcal{T}_{[t,T]}} \mathbb{E}[e^{-r(T-t)} \varphi(S_\tau^{t,x})]$, where $\mathcal{T}_{[t,T]}$ is the set of $(\mathcal{F}_\theta)_{0 \leq \theta \leq T}$ -adapted stopping times τ taking values in [t,T] a.s., and the payoff function is $\varphi(x) = (K - x)_+$. It is possible to show that $v(t,x) = V(T-t,x)$ satisfies the following nonlinear PDE :

$$-min(v_t + \mathcal{A}v, v - \varphi(x)) = 0,\ x \geq 0, t \in (0,T),$$

$$\begin{gathered} v(t, Xmin) = v_l(t) \equiv K - Xmin,\ t \in (0,T), \\ v(t, Xmax) = v_r(t) \equiv 0,\ t \in (0,T), \\ v(0,x) = \varphi(x),\ x \geq 0 \end{gathered}$$

with

$$\mathcal{A}v := -\tfrac{1}{2}\sigma^2 x^2 v_{xx} - rxv_x + rv,$$

We are going to compute an approximation of the exact solution using different approaches, detailed in the following parts. We will adopt the usual notations :
mesh $x_j = Xmin + jh, j = 1..I$, $h = (Xmax - Xmin)/(I+1)$ and $t_n = n\delta t$, $0 \leq n \leq N$, $\delta t = T/N$. We look for U_j^n, an approximation of $u(t_n, x_j)$. We choose to work with the unknown vector of $\mathbb{R}^I$:

$$U^n := \begin{pmatrix} U_1^n \\ \vdots \\ U_I^n \end{pmatrix}$$

For next parts, we will take as values for our global variables like the following :
$K = 100, r = 0.1, sigma = 0.2, T = 1, Xmin = 0, Xmax = 200.$

1

Development

1.1 Explicit Euler Forward Scheme

Before starting approximation part, we want to clarify that we had used the solution approximated from the BDF2 scheme for the price of American put option at t=0 and for x=117.1417 as the exact solution with fine mesh (2560,2560), since it's difficult to calculate an analytic solution for American put option, to estimate errors of the following schemes. We want to mention also that x=117.1417 belongs to the mesh of BDF2 (I=N=2560).

$$v(t,x) = \sum_{n=1}^{M}\sum_{i=1}^{N}(v_{h,k}^n)_i 1_{(xi-\frac{h}{2},xi+\frac{h}{2})}(x) * 1_{(n-1)k,nk)}(t)$$

Let us consider the explicit Euler Forward Scheme (EE) with centred approximation [1]:

$$\partial_x u(t_n, x_j) \simeq \frac{U_{j+1}^n - U_{j-1}^n}{2h}$$

We obtain :

$$\begin{gathered} min(\frac{U_j^{n+1}-U_j^n}{\delta t} + \frac{\sigma^2}{2}x_j^2\frac{-U_{j-1}^n+UP_j^n-U_{j+1}^n}{h^2} \text{ - } rx_j\frac{U_{j+1}^n-U_{j-1}^n}{2h} + rU_j^n \ , \ U_j^{n+1} \text{ - } \varphi(x_j)) \\ = 0 \ , \ 1 \leq j \leq I, \\ U_0^n = v_l(t_n), \\ U_{I+1}^n = v_r(t_n), \end{gathered}$$

for $n = 0..N-1$. The scheme is initialized with $U_j^0 = \varphi(x_j)$. We denote by A The discretization matrix associated to the operator $\mathcal{A}$, of size I, and $q(t) \in \mathbb{R}^I$, such that

$$(AP + q(t))_j := \frac{\sigma^2}{2} x_j^2 \frac{-Pj-1+2Pj-P_{j+1}}{h^2} \text{ - } rx_j \frac{P_{j+1}-P_{j-1}}{2h} + rP_j,\ 1 \leq j \leq I$$
$$= -(\alpha_j - \beta_j)P_{j-1} + (2\alpha_j + r)P_j \text{ - } (\alpha_j + \beta_j)P_{j+1},\ 1 \leq j \leq I$$

with $\alpha_j = \frac{\sigma^2}{2}\frac{x_j^2}{h^2}$ and $\beta_j = \frac{rx_j^2}{2h}$. We recall that A is the tridiagonal matrix

$$tridiag(-(\alpha_j - \beta_j), 2\alpha_j + r, -(\alpha_j + \beta_j))$$

and

$$q(t) := \begin{pmatrix} (-\alpha_1 + \beta_1)\mathrm{v}_l(t) \\ 0 \\ \vdots \\ 0 \\ (-\alpha_I - \beta_I)\mathrm{v}_r(t) \end{pmatrix}$$

Let also g be the vector of $\mathbb{R}^I$ with components $g_j := \varphi(x_j)$. We finally obtain the following equivalent form of the scheme (EE) in $\mathbb{R}^I$:

$$min(\frac{U^{n+1}-U^n}{\delta t} + AU^n + q(t_n),\ U^{n+1} \text{ - } g) = 0\ ,\ n = 0..N-1,$$
$$U^0 = g,$$

The main iteration can be written

$$U_i^{n+1} = max(U_i^n \text{ - } \delta t(AU^n q(t_n))_i,\ g),\ 1 \leq i \leq I$$

Or, in vector form

$$U^{n+1} = max(U^n \text{ - } \delta t(AU^n q(t_n)),\ g)$$

Approximation

As we mentioned in section 1, we have estimated the value of American pit option at t=0 for x=117.1417 which exists in the mesh of our reference scheme BDF2.
We had tested the program with many meshes, we put our results in the next table, where we put the CFL's value in addition to our comments about the stability of the solution.

Mesh	Error 'inf'	CFL	Stability
J N	Error		
30 30	9,60E-02	1.281	Stable
40 40	6,70E-03	1,681	Stable
50 50	4,98E-02	2.081	unstable
60 60	2,57E-01	2.481	unstable
70 70	1.12105	2.881	unstable
50 20	1.12105	5,202	unstable

Figure 1.1: Errors between the approximated and the reference solution using the infinity norm at t=0.

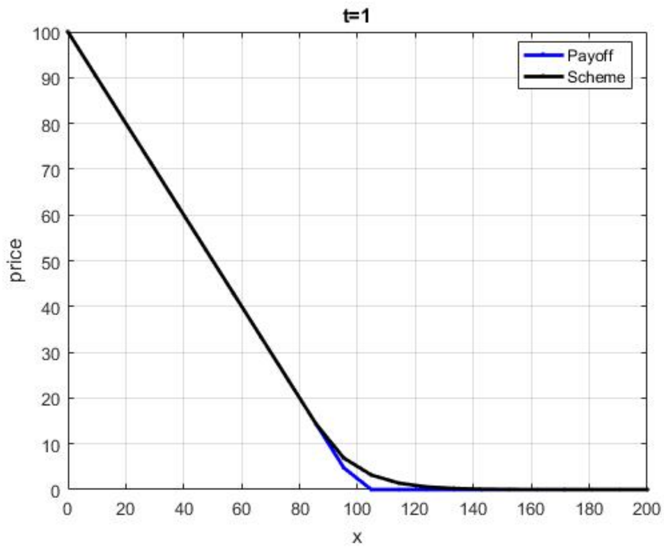

Figure 1.2: The price of American put option at t=0 for terminal T=1 with I=40 and N=40.

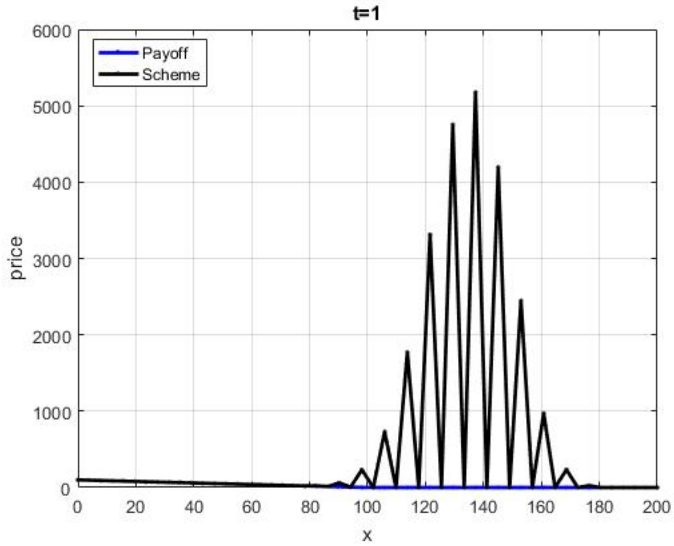

Figure 1.3: The price of American put option at t=0 for terminal T=1 with I=50 and N=20.

We note that the program gives a solution with a stable behavior when CFL is small, which corresponds with our theoretical demonstration. In contrary, it gives a solution with an unstable behavior, when CFL is big (unsatisfied), so our program doesn't respect the condition imposed on CFL. Finally, we can admit that the explicit scheme is convergent ant stable if and only if CFL is small.

1.2 Implicit Euler Scheme

For stability reasons, we turn on to the time-implicit Euler scheme, which takes the following form:

$$min(\tfrac{U^{n+1}-U^n}{\delta t} + AU^{n+1} + q(t_{n+1}),\ U^{n+1} \text{ - } g) = 0\ ,\ n = 0..N-1,$$
$$U^0 = g,$$

Let us define $B := I_d + \delta t A$ and $b := U^n - \delta t q(t_{n+1})$.
For each n, one must solve a solution x $\in \mathrm{R}^I$ of the following non-linear system

$$min(Bx - b, x - g) = 0\ ,\ \text{in } \mathbb{R}^I.$$

For the treatment of the obstacle term for the implicit Euler scheme, we had tested projected UL method, Newton's method and PSOR method.

1.2.1 Projected UL method

The idea is to decompose B in this form : $B = UL$, L is a lower triangular matrix and U is an upper triangular matrix with $U_{ii} = 1$.
To solve $min(Bx - b, x - g) = 0$, we use the equivalence:

$$min(ULx - b, x - g) = 0 \Longleftrightarrow min(Lx - U^{-1}b, x - g) = 0$$

The right hand side of this equivalence has an explicit solution given by
(i) solve $c = U^{-1}b$; **upwind** algorithm.
(ii) solve $min(Lx - c, x - g) = 0$; **downwind** algorithm.

Approximation

We had tested the program with many meshes, we put our results in the next table, where we put the CFL's value in addition to our comments about the stability of the solution.

Mesh		Error 'inf'	CFL	Stability
J	N	Error		
80	80	1,04E-01	3.281	Stable
160	160	3,37E-02	6.480	Stable
320	320	1,14E-03	12.880	Stable
640	640	1,30E-02	25.680	Stable
1280	1280	4,15E-03	51.280	Stable
2560	2560	2,20E-04	102.480	Stable

Figure 1.4: Errors between the approximated and the reference solution at t=0.

Mesh		Error 'inf'	CFL	Stability
J	N	Error		
80	8	5,52E-02	32.805	Stable
160	16	8,26E-03	64.803	Stable
320	32	1,44E-02	128.801	Stable
640	64	5,87E-03	256.801	Stable
1280	128	4,90E-04	512.800	Stable
2560	256	2,11E-03	1024.800	Stable

Figure 1.5: Errors between the approximated and the reference solution at t=0, with higher CFL number.

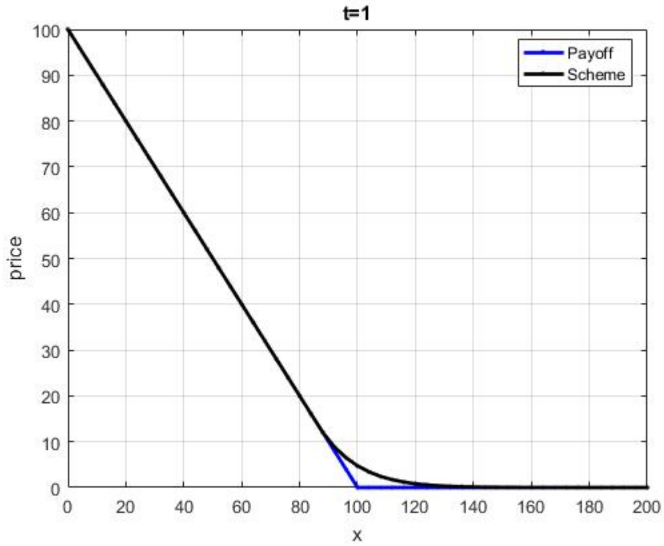

Figure 1.6: The price of American put option at t=0 for terminal T=1 with I=160 and N=16.

We note that the program gives a stable solution which has converged well to the reference solution despite that the CFL is clearly not small. We can affirm that in the implicit scheme, no need of the CFL condition.

1.2.2 Newton's method

We are going to use a Newton type algorithm to solve $F(x) = 0$, with $F(x) := min(Bx - b, x - g) = 0$. For a given $x^0 \in \mathbb{R}^I$, we look for x such that $x^{k+1} = x^k - F'(x^k)^{-1}F(x^k)$, $k \geq 0$.

Approximation

We had made the same test as with UL method and we obtained the same results.

1.2.3 PSOR method

The idea is to decompose B in this form : $B = U + L$, L is a lower triangular matrix with $L_{ii} = B_{ii}$ and U_{ii} is an upper triangular matrix with
$U_{ii} = 0$.
To solve $min(Bx - b, x - g) = 0$, we use the equivalence:

$$min((U+L)x-b, x-g) = 0 \Longleftrightarrow min(Lx-(b-Ux), x-g) = 0$$

x will be determined by the projected downwind algorithm, we denote $x = q_L(b-Ux)$. The PSOR method is given by this algorithm:

For a given $x^0 \in \mathbb{R}^I$, $x^{k+1} = q_L(b-Ux^k)$, $k \geq 0$.

We had made the same test as before and we obtained same the results.

1.3 Crank-Nicolson Scheme ($\theta = \frac{1}{2}$)

The Crank-Nicolson scheme takes the following form:

$$min(\frac{U^{n+1}-U^n}{\delta t} + \frac{1}{2}(AU^{n+1} + q(t_{n+1})) + \frac{1}{2}(AU^n + q(t_n)),\ U^{n+1} - g) = 0\ ,$$
$$n = 0..N-1,$$
$$U^0 = g,$$

We are facing again an obstacle problem $min(Bx-b, x-g) = 0$. For the treatment of the obstacle term for the Crank-Nicolson Scheme, we had used the Newton method.

1.3.1 Approximation

As before, we had tested the program with many meshes, we put our results in the next table, where we put the CFL's value in addition to our comments about the stability of the solution.

Mesh		Error 'inf'	CFL	Stability
J	N	Error		
80	80	0.11091	3.281	Stable
160	160	3,69E-02	6.480	Stable
320	320	5,20E-04	12.880	Stable
640	640	1,38E-02	25.680	Stable
1280	1280	4,60E-03	51.280	Stable
2560	2560	1,00E-05	102.480	Stable

Figure 1.7: Errors between the approximated and the reference solution using the infinity norm at t=0.

Mesh		Error 'inf'	CFL	Stability
J	N	Error		
80	8	1,19E-01	32.805	Stable
160	16	3,60E-02	64.803	Stable
320	32	1,30E-04	128.801	Stable
640	64	1,36E-02	256.801	Stable
1280	128	4,50E-03	512.800	Stable
2560	256	5,00E-05	1024.800	Stable

Figure 1.8: Errors between the approximated and the reference solution at t=0, with higher CFL number.

We note that the program gives a stable solution which has converged well to the reference solution.

Finally we conclude that for the explicit scheme, with respect to the condition imposed on CFL, the scheme is stable and solution is convergent. On the contrary, for the implicit scheme and the crank nicolson, there is usually stability and convergence.

1.4 Implicit Backward Difference Scheme(BDF2)

In this part, we will study an Implicit Backward Euler Scheme [2][3]

$$min((I_d + \tfrac{2}{3}\delta tA)U^{n+1} - \tfrac{4}{3}U^n + \tfrac{1}{3}U^{n-1} + \tfrac{2}{3}\delta tq(t_{n+1}),\, U^{n+1} - g) = 0$$

We denote $B{:=}(I_d + \frac{2}{3}\delta tA)$, $b{:=}\frac{4}{3}U^n - \frac{1}{3}U^{n-1} - \frac{2}{3}\delta tq(t_{n+1})$, the problem is to solve for x in $\mathbb{R}^I$:

$$min(Bx - b, x - g) = 0$$

For the treatment of the obstacle term, we had used the Newton method. Or we had used the solution from this scheme (I=N=2560) at t=0 and for $x = 117.1417$ like exact solution, we go now to study more this scheme. We had tested the program with many meshes less fine, we put our results in the next table, where we put the CFL's value in addition to our comments about the stability of the solution.

Mesh		Error 'inf'	CFL	Stability
J	N	Error		
80	80	0.11041	3.281	Stable
160	160	3,67E-02	6.480	Stable
320	320	4,40E-04	12.880	Stable
640	640	1,38E-02	25.680	Stable
1280	1280	4,60E-03	51.280	Stable
2560	2560	0,00E+00	102.480	Stable

Figure 1.9: Errors between the approximated solution in BDF2 scheme and the reference solution (I=2560, N=2560) at t=0.

Mesh		Error 'inf'	CFL	Stability
J	N	Error		
80	8	9,90E-02	32.805	Stable
160	16	3,30E-02	64.803	Stable
320	32	1,15E-03	128.801	Stable
640	64	1,31E-02	256.801	Stable
1280	128	4,26E-03	512.800	Stable
2560	256	1,50E-04	1024.800	Stable

Figure 1.10: Errors between the approximated solution in BDF2 scheme and the reference solution (I=2560, N=2560) at t=0, with higher CFL.

We note that the scheme BDF2 gives a stable solution which has converged well to the reference solution despite that the CFL is clearly not small. We can affirm that in the BDF2 scheme, no need of the CFL condition.

1.5 Comparison between schemes

Mesh		Implicit Newton	Crank Nicolson	BDF2	CFL	Stability
J	N	Error	Error	Error		
80	80	1,04E-01	0.11091	0.11041	3.281	Stable
160	160	3,37E-02	3,69E-02	3,67E-02	6.480	Stable
320	320	1,14E-03	5,20E-04	4,40E-04	12.880	Stable
640	640	1,30E-02	1,38E-02	1,38E-02	25.680	Stable
1280	1280	4,15E-03	4,60E-03	4,60E-03	51.280	Stable
2560	2560	2,20E-04	1,00E-05	0,00E+00	102.480	Stable

Figure 1.11: Errors between the approximated solution in different schemes and the reference solution at t=0.

Mesh		Implicit Newton	Crank Nicolson	BDF2	CFL	Stability
J	N	Error	Error	Error		
80	8	5,52E-02	1,19E-01	9,90E-02	32.805	Stable
160	16	8,26E-03	3,60E-02	3,30E-02	64.803	Stable
320	32	1,44E-02	1,30E-04	1,15E-03	128.801	Stable
640	64	5,87E-03	1,36E-02	1,31E-02	256.801	Stable
1280	128	4,90E-04	4,50E-03	4,26E-03	512.800	Stable
2560	256	2,11E-03	5,00E-05	1,50E-04	1024.800	Stable

Figure 1.12: Errors between the approximated solution in different schemes and the reference solution at t=0, with higher CFL.

The two figures show robustness of the BDF2 scheme with higher CFL numbers mainly, and also an improvement of the convergence with respect to the CN schemes, and the implicit scheme too in the case of high CFL mainly.

Additionally as like we see in the article of D.Lamberton and B.Lapeyeyre, the Crank Nicolson Scheme is conditionally convergent, it has an unstable behavior if $(stepintime/(stepinspace)^2)$ isn't small enough, but we do not know much about the convergence of the CN-scheme for finite differences for the American option if $(stepintime/(stepinspace)^2)$ is big.

1.6 Problems

At the first, we had faced a problem with the choice of values of Xmin, Xmax and K because of their importance, mainly K, the price "strike", which influences mostly on our results.

Secondly and the most important, we have felt that the problem has lost a lot of its rigor when we considered an approximated solution like an exact solution to make our comparison even if we had taken a fine mesh.

Bibliography

[1] Finite difference schemes for american options. (https://www.ljll.math.upmc.fr/∼bokanowski/enseignement/2017/FQ301/FQ301_2017.html)

[2] O.Bokanowski, S.Maroso, H.Zidani. "Some convergence results for Howard's algorithm,", Preprint Inria, 2007. (http://hal.inria.fr/inria-00179549/fr/)

[3] O. Bokanowski, K. Debrabant,. "High order finite difference schemes for some diffusion-obstacle problems" (2015).