Project : Option Pricing With Finite Difference And Finite Element Methods

Alexandre Goldman, Baptiste Borne.

Introduction

In this project, we will study different ways of pricing options with finite difference and finite element methods. We will then implement those numerical methods in Python and analyse the results.

Références

[1] Finite Difference Schemes for Pricing of European and American Option, Margarida Mirador Fernandes

[2] Méthodes numériques pour le Pricing d'Options, Didier Auroux

[3] Options, Futures et autres actif dérivés, J.Hull

[4] Real options pricing by the finite element method, A. Andalaft-Chacur, M. Montaz Ali,, J. González Salazar

1 Finite Difference Method

1.1 Introduction

For this method, we will first price European Options with two different changes of variables :

- One to come back to a heat equation for a given function with new coordinates, functions of the coordinates of the inital option.
- A simpler change of variable, called log-transformation, simplifying the equation.

We will then adapt this method to price Barrier Option.

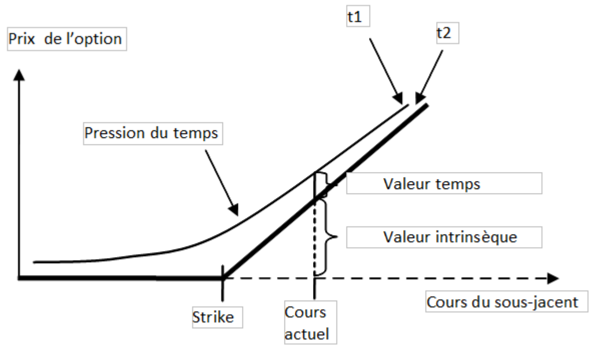

Figure 1.1: Call Option - Price decomposition

Définition 1. (Black-Scholes Equation) Under the asumptions of Black-Scholes, the behabiour of European options can be described through the following equation :

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

where V(S,t) is the price at t when the underlying is worth S. ($t \in [0,T]$, $S \in [0,\infty]$)

- Final Condition for a European Call :

$$C(S,T) = max(S-K,0), \forall S \in [0,\infty]$$

- Boundary Conditions for a European Call :

$$\begin{aligned} C(0,t) &= 0 \\ \lim_{S\to\infty} C(S,t) &= S-K \end{aligned}$$

1.2 European Option

1.2.1 Resolution based on Reduction to Heat Equation

1.2.1.1 Changes of variables Let's do a first change of variable to eliminate the two coefficients in S and S^2 multiplying the first and second order spatial operators : $S = Ke^x, t = T - \frac{\tau}{\frac{1}{2}\sigma^2}, C = Kv(x,\tau)$. Let $k = \frac{r}{\frac{1}{2}\sigma^2}$, the equation becomes, after some easy computations :

$$-\frac{\partial v}{\partial \tau} + \frac{\partial^2 v}{\partial x^2} + (k-1)\frac{\partial v}{\partial x} - kv = 0 \ (1)$$

The initial condition becomes :

$$v(x,0) = max(e^x - 1, 0) = C(x, \tau = 0)$$

The new time interval considered is then $\left[0, \frac{1}{2}\sigma^2 T\right]$ and the new space interval considered is $[-\infty; +\infty]$. Let's now do a second change of variables in order to eliminate the spatial operator of order 0 and 1: $v = e^{\alpha x + \beta\tau} u(x, \tau)$ α and β to be determined. We have:

$$\begin{aligned}
\frac{\partial v}{\partial \tau} &= \beta e^{\alpha x+\beta\tau} u + e^{\alpha x+\beta\tau}\frac{\partial u}{\partial \tau} \\
\frac{\partial v}{\partial x} &= \alpha e^{\alpha x+\beta\tau} u + e^{\alpha x+\beta\tau}\frac{\partial u}{\partial x} \\
\frac{\partial^2 v}{\partial x^2} &= \alpha^2 e^{\alpha x+\beta\tau} u + \alpha e^{\alpha x+\beta\tau}\frac{\partial u}{\partial x} + \alpha e^{\alpha x+\beta\tau}\frac{\partial u}{\partial x} + e^{\alpha x+\beta\tau}\frac{\partial^2 u}{\partial x^2}
\end{aligned}$$

By replacing the previous equations in the equation (1) and dividing by the common factor of $e^{ax+\beta\tau}$ we get the following equation on u(x,τ):

$$-\beta u - \frac{\partial u}{\partial \tau} + \alpha^2 u + 2\alpha\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + (k-1)(\alpha u + \frac{\partial u}{\partial x}) - ku = 0$$

$$-\frac{\partial u}{\partial \tau} + \frac{\partial^2 u}{\partial x^2} + [2\alpha + (k-1)]\frac{\partial u}{\partial x} + [\alpha^2 + (k-1)\alpha - k - \beta]u$$

We choose $\alpha = -\frac{k-1}{2}$ so that the coefficient is 0 and choose $\beta = \alpha^2 + (k-1)\alpha - k = \frac{-(k+1)^2}{4}$ so the u coefficient is likewise 0. The initial condition becomes :

$$u(x, \tau = 0) = max(e^{x-\alpha x} - e^{-\alpha x}, 0) = max(e^{\frac{(k+1)}{2}x} - e^{\frac{(k-1)}{2}x}, 0)$$

With this choice, the equation we get is reduced to :

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} \ (2)$$

1.2.1.2 Implementation of Explicit Scheme The Explicit Scheme uses for the computation of the derivatives of u with respect to the time a finite difference off center to the right. We get the following discretised equation for equation (2)

$$\frac{u_n^{m+1} - u_n^m}{\delta t} = \frac{u_{n-1}^m - 2u_n^m + u_{n+1}^m}{\delta x^2} \ (3)$$

For the stability of the scheme we need $a = \frac{\delta t}{\delta x^2} < \frac{1}{2}$ and we rewrite (3) as $u_n^{m+1} = a u_{n-1}^m + (1-2\alpha)u_n^m + \alpha u_{n+1}^m$.which hold for all the value of u exept for the 2 values on the boundaries.

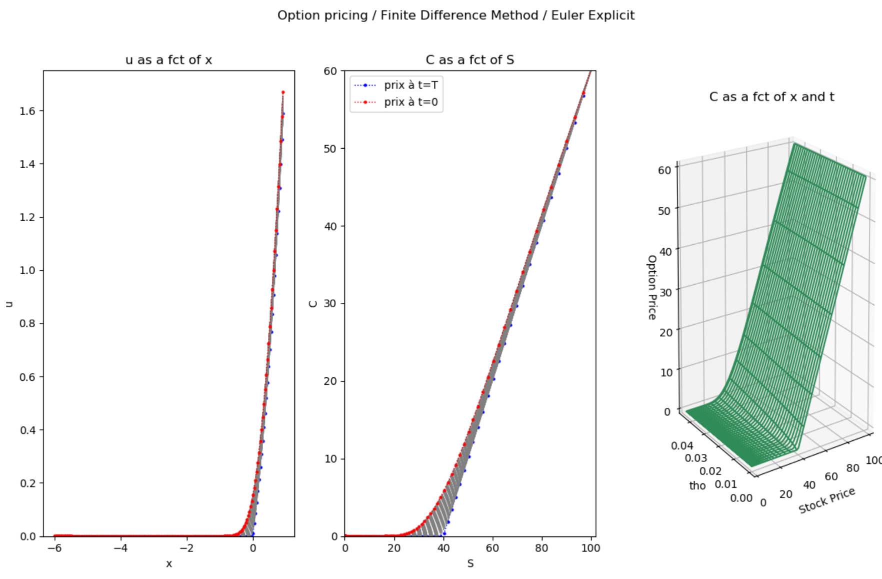

FIGURE 1.2 : FDM - Option pricing using the Heat Equation - Explicit

1.2.1.3 Implementation of Implicit Scheme The Implicit scheme uses for the computation of the derivatives of u with respect to the time a finite difference off center to the left We get the following discretised equation for equation (2): $\frac{u_n^m - u_n^{m-1}}{\delta t} = \frac{u_{n-1}^m - 2u_n^m + u_{n+1}^m}{\delta x^2}$ which gives the final discretised form :

$$\frac{u_n^{m+1} - u_n^m}{\delta t} = \frac{u_{n-1}^{m+1} - 2u_n^{m+1} + u_{n+1}^{m+1}}{\delta x^2} \quad (4)$$

We rewrite (4) with $a = \frac{\delta t}{\delta x^2}$ to get

$$-au_{n-1}^{m+1} + (1+2a)u_n^{m+1} - au_{n+1}^{m+1} = u_n^m.$$

which which hold for all the value of u exept for the 2 values on the boundaries. Thus we rewrite that solution with matrices. We use for the Implicit Scheme the following discretized equation :

- $$\begin{pmatrix} 1+2a & -a & 0 & \dots & 0 \\ -a & 1+2a & -a & & 0 \\ 0 & -a & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & -a \\ 0 & 0 & & -a & 1+2a \end{pmatrix} \begin{pmatrix} u_{N-1}^{n+1} \\ u_{N-2}^{n+1} \\ \vdots \\ u_2^{n+1} \\ u_1^{n+1} \end{pmatrix} = \begin{pmatrix} u_{N-1}^{n} \\ u_{N-2}^{n} \\ \vdots \\ u_2^{n} \\ u_1^{n} \end{pmatrix} + \begin{pmatrix} au_N^{n+1} \\ 0 \\ \vdots \\ 0 \\ au_0^{n+1} \end{pmatrix}$$

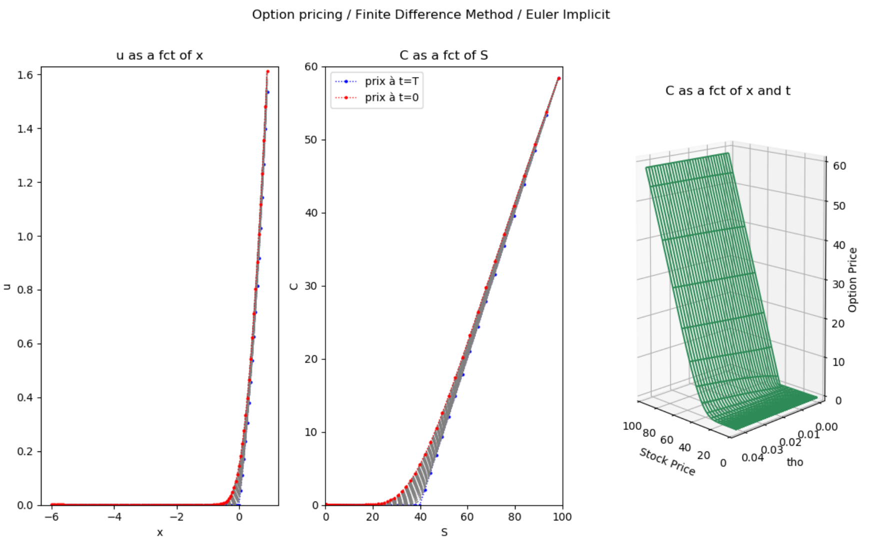

FIGURE 1.3 : FDM - Option Pricing using the Heat Equation - Implicit

1.2.1.4 Implementation of Cranck-Nicolson Scheme

$$\frac{u_n^m - u_n^{m-1}}{\delta t} = \frac{1}{2}\left(\frac{u_{n-1}^m - 2u_n^m + u_{n+1}^m}{\delta x^2} + \frac{u_{n-1}^{m+1} - 2u_n^{m+1} + u_{n+1}^{m+1}}{\delta x^2}\right)(5)$$

The error of that scheme is in $O(\delta\tau^2 + \delta x^2)$, which doesn't require any particular asumptions on $\frac{\delta t}{\delta x^2}$to guarantee the precision. We rewrite (5) with $a = \frac{\delta t}{\delta x^2}$ to get

$$u_n^{m+1} - \frac{1}{2}a(u_{n-1}^{m+1} - 2u_n^{m+1} + u_{n+1}^{m+1}) = u_n^m + \frac{1}{2}a(u_{n-1}^m - 2u_n^m + u_{n+1}^m)$$

which which hold for all the value of u exept for the 2 values on the boundaries. Thus we rewrite that solution with matrices.

$$\begin{pmatrix} 1+a & \frac{-a}{2} & 0 & ... & 0 & 0 \\ \frac{-a}{2} & 1+a & \frac{-a}{2} & 0 & ... & 0 \\ 0 & \frac{-a}{2} & 1+a & \frac{-a}{2} & ... & ... \\ ... & 0 & ... & ... & ... & 0 \\ ... & ... & ... & \frac{-a}{2} & 1+a & \frac{-a}{2} \\ 0 & 0 & ... & 0 & \frac{-a}{2} & 1+a \end{pmatrix} \begin{pmatrix} u_{N-1}^{m+1} \\ ... \\ ... \\ ... \\ ... \\ u_1^{m+1} \end{pmatrix} = \begin{pmatrix} f_{N-1}^m \\ ... \\ ... \\ ... \\ ... \\ f_1^m \end{pmatrix} - \begin{pmatrix} \frac{a}{2}u_{N-1}^{m+1} \\ 0 \\ ... \\ ... \\ 0 \\ \frac{a}{2}u_0^{m+1} \end{pmatrix}$$

where $f_n^m = (1-a)u_n^m + \frac{1}{2}a(u_{n-1}^m + u_{n+1}^m)$

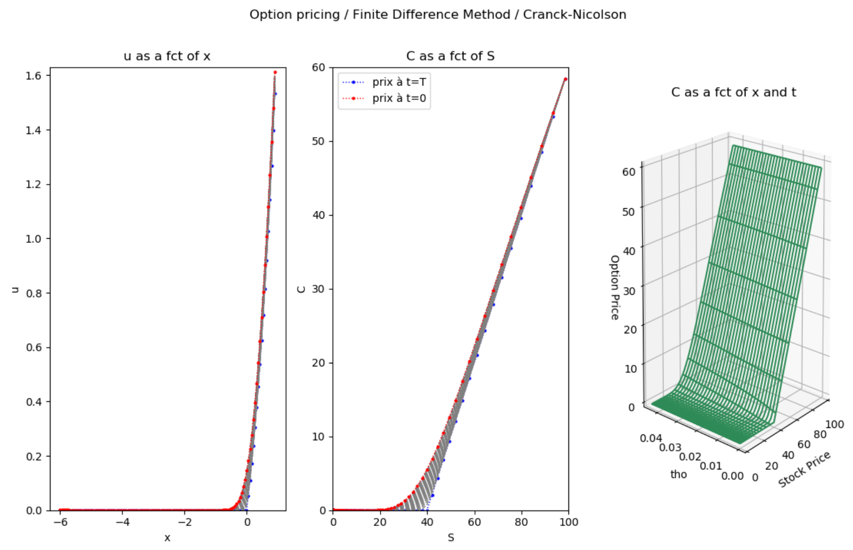

FIGURE 1.4 : FDM - Option Pricing using the Heat Equation - Cranck-Nicolson

1.2.1.5 Comparison of convergence in space In order to compare the convergence in space of the three schemes, we decided to compute the average of squared error, relatively to the exact solution of Black-Scholes, for increasing values of the number of spacing points (i.e. decreasing space mesh size). Then, if we denote $C^0 = (C_i^0)_{i \in [0,N]}$ the vector containing the spatial solution of our scheme at time t=0, $s = (s_i)_{i \in [[o,N]}$ the space mesh and $BS(t, S)$ the exact solution of Black-Scholes at point (t, S), we can define our relative error as follows :

$$RE(N) := \frac{1}{N} \sum_{i=0}^{N} [C_i^0 - BS(t=0, S=s_i)]^2 \text{ , for } N \text{ in } \mathbb{N}$$

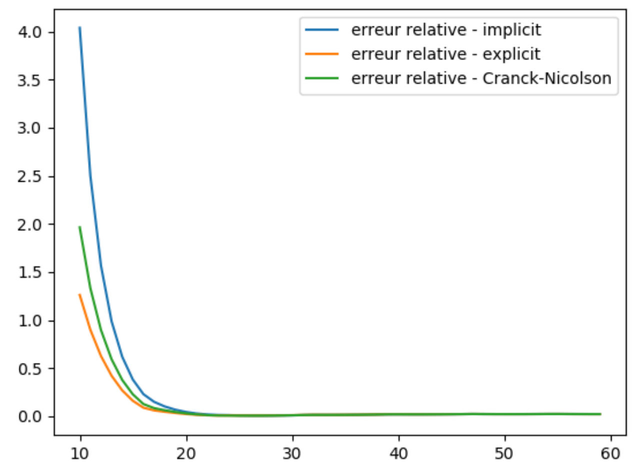

FIGURE 1.5 : Relative error for the 3 Schemes using the Heat Equation

1.2.2 Resolution based on z=log(S) transformation

Using this new change of variable, we have that $v(z,t) = C(S,t)$. Let's compute the first and second order space derivatives:

$$\frac{\partial C}{\partial S} = \frac{\partial z}{\partial S}\frac{\partial v}{\partial z} = \frac{1}{S}\frac{\partial v}{\partial z}$$

$$\frac{\partial^2 C}{\partial S^2} = \frac{\partial z}{\partial S}\frac{\partial v}{\partial z} = \frac{1}{S^2}\frac{\partial^2 v}{\partial z^2} - \frac{1}{S^2}\frac{\partial v}{\partial z}$$

The Black-Scholes differential equation $\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0$ then becomes :

$$\frac{\partial \nu}{\partial t} + rS(\frac{1}{S}\frac{\partial v}{\partial z}) + \frac{1}{2}\sigma^2 S^2(\frac{1}{S^2}\frac{\partial^2 v}{\partial z^2} - \frac{1}{S^2}\frac{\partial v}{\partial z}) - r\nu = 0$$

$$\text{i.e. } \frac{\partial \nu}{\partial t} + (r - \frac{1}{2}\sigma^2)\frac{\partial v}{\partial z} + \frac{1}{2}\sigma^2\frac{\partial^2 v}{\partial z^2} - rv = 0 \ (6)$$

Let's check the ..... in the case of explicit scheme. Let's denote $u_j^n = A(k)^n e^{2i\pi k x_j}$ where $x_j = j\Delta x$, then replace in the equation above:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + (r - \frac{\sigma^2}{2})\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \frac{\sigma^2}{2}\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x} - ru_j^n = 0$$

$$\frac{A(k)^{n+1}e^{2i\pi kj\Delta x}-A(k)^{n}e^{2i\pi kj\Delta x}}{\Delta t}+(r-\frac{\sigma^2}{2})\frac{A(k)^{n}e^{2i\pi k(j+1)\Delta x}-A(k)^{n}e^{2i\pi k(j-1)\Delta x}}{2\Delta x}+\frac{\sigma^2}{2}\,A(k)^{n}e^{2i\pi k(j+1)\Delta x}$$

Let's divide by $A(k)^n e^{2i\pi kj\Delta x}$:

$$\frac{A(k)-1}{\Delta t}+(r-\frac{\sigma^2}{2})\frac{e^{2i\pi k\Delta x}-e^{-2i\pi k\Delta x}}{2\Delta x}+\frac{\sigma^2}{2}\frac{e^{2i\pi k\Delta x}-2+e^{-2i\pi k(\Delta x}}{\Delta x}-r=0$$

$$A(k)=1+\frac{\Delta t}{\Delta x}$$

Let's use a Crank-Nicolson Scheme for implementation:

$$\begin{aligned}
\frac{\partial v}{\partial t} &= \frac{v_n^{m+1}-v_n^m}{\delta t}\\
\frac{\partial v}{\partial z} &= \frac{1}{2}(\frac{u_{n+1}^{m+1}-u_{n-1}^{m+1}}{2\delta x}+\frac{u_{n+1}^{m}-u_{n-1}^{m}}{2\delta x})\\
\frac{\partial^2 v}{\partial z^2} &= \frac{1}{2}(\frac{v_{n-1}^{m}-2v_n^m+v_{n+1}^{m}}{\delta x^2}+\frac{v_{n-1}^{m+1}-2v_n^{m+1}+v_{n+1}^{m+1}}{\delta x^2})
\end{aligned}$$

Replacing the approximations of the derivatives in the equation (6) we obtain the Crank-Nicolson scheme.

$$av_{n+1}^{m+1}+dv_n^{m+1}+cv_{n-1}^{m+1}=a1v_{n+1}^{m}+d1v_n^{m}+c1v_{n-1}^{m}$$

$$\text{where :}\begin{cases}
a=\frac{1}{4}(\frac{\sigma^2}{\delta x^2}+\frac{(r-\frac{\sigma^2}{2})}{\delta x}) & a1=-a\\
d=\frac{1}{\delta t}-\frac{1}{2}\frac{\sigma^2}{\delta x^2} & d1=\frac{1}{\delta t}+r+\frac{1}{2}\frac{\sigma^2}{\delta x^2}\\
c=\frac{1}{4}(\frac{\sigma^2}{\delta x^2}-\frac{(r-\frac{\sigma^2}{2})}{\delta x}) & c1=-c
\end{cases}$$

In matrix form, if we denote $f_n^{m+1}=av_{n+1}^{m+1}+dv_n^{m+1}+cv_{n-1}^{m+1}$, we obtain :

$$\begin{pmatrix}
d1 & a1 & 0 & ... & 0 & 0\\
c1 & d1 & a1 & 0 & ... & 0\\
0 & c1 & d1 & a1 & ... & ...\\
... & 0 & ... & ... & ... & 0\\
... & ... & ... & c1 & d1 & a1\\
0 & 0 & ... & 0 & c1 & d1
\end{pmatrix}
\begin{pmatrix} v_{N-1}^{m+1}\\ ...\\ ...\\ ...\\ ...\\ v_1^{m+1}\end{pmatrix}
=
\begin{pmatrix} f_{N-1}^{m}\\ ...\\ ...\\ ...\\ ...\\ f_1^{m}\end{pmatrix}
-
\begin{pmatrix} c1u_N^m\\ 0\\ ...\\ ...\\ 0\\ a1u_0^m\end{pmatrix}$$

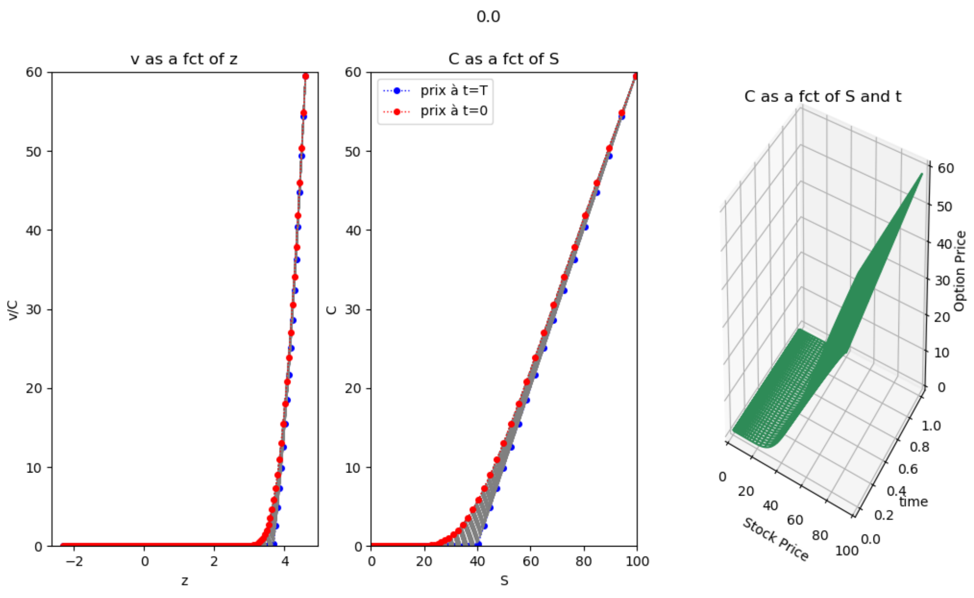

Figure 1.6: FDM - Option Pricing using the log trasnformation - Cranck-Nicolson

1.3 Barrier Option

A barrier option can be American or European. When they are european what distinguishs them from vanilla options is the fact that a barrier B is introduced. We will focus on call european option. When the asset value S reaches the barrier, the following results can be seen: If it is a Knock-In option, the option is enabled. If it is a Knock-out option, the option ceases to exist. We will denote by K(S,t) the value of a barrier option and as usual V(S,t) is the value of the respective vanilla option

$Up - Out$	$Up - In$
$K(S,T) = V(S,T)1\{sup_{0<t<T} St < B\}$	$K(S,T) = V(S,T)1\{sup_{0<t<T} St >= B\}$
$K(B,t) = 0$	$K(B,t) = V(B,t)$
$K(0,t) = V(0,t)$	$K(0,t) = V(0,t)$

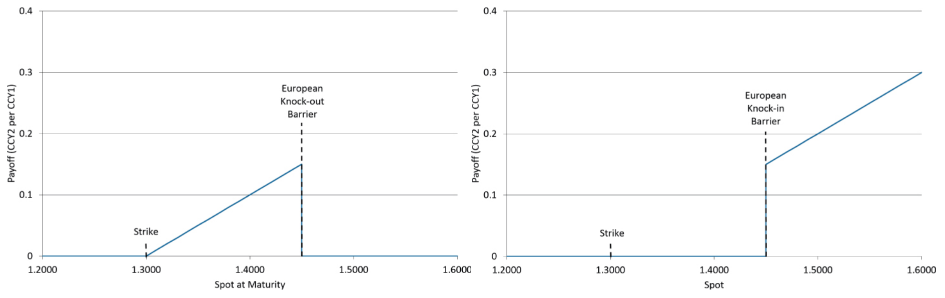

Figure 1.7: Payoff of Up and Out and Up and In Barrier Options

We then applied the three schemes used above to price Barrier Options. It is quite easy to adapt the methods used for vanilla option, putting one of the boundary condition on the value of the barrier. Let's first take the example of an up-and-out call option, i.e. a call option that "desactivates" when a certain barrier is reached, therefore having no more value as soon as this barrier is reached. Then, we only need to price the option assuming the barrier hasn't been reached yet. To do that, one can simply put the boundary condition on the value of the barrier, zero in this case.

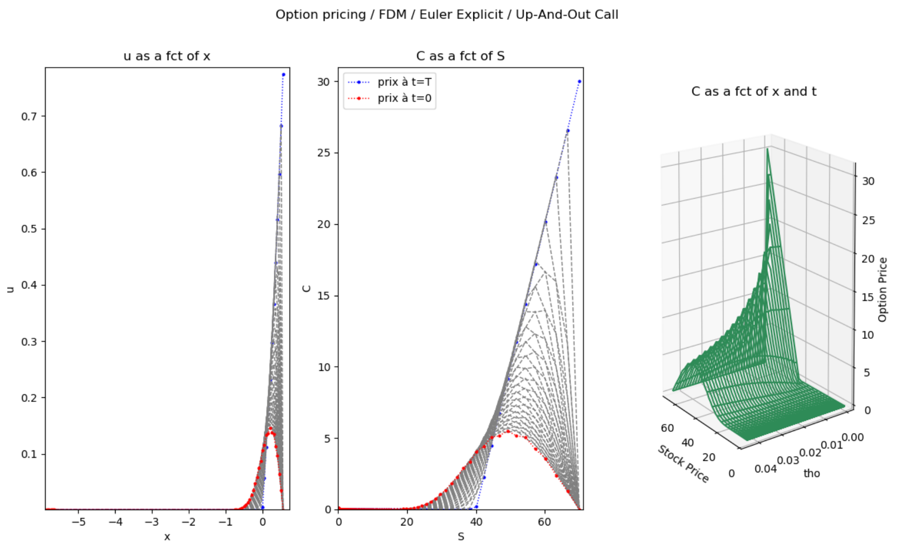

Figure 1.8 : FDM - Up and Out Option Pricing using the Heat Eqt - Explicit

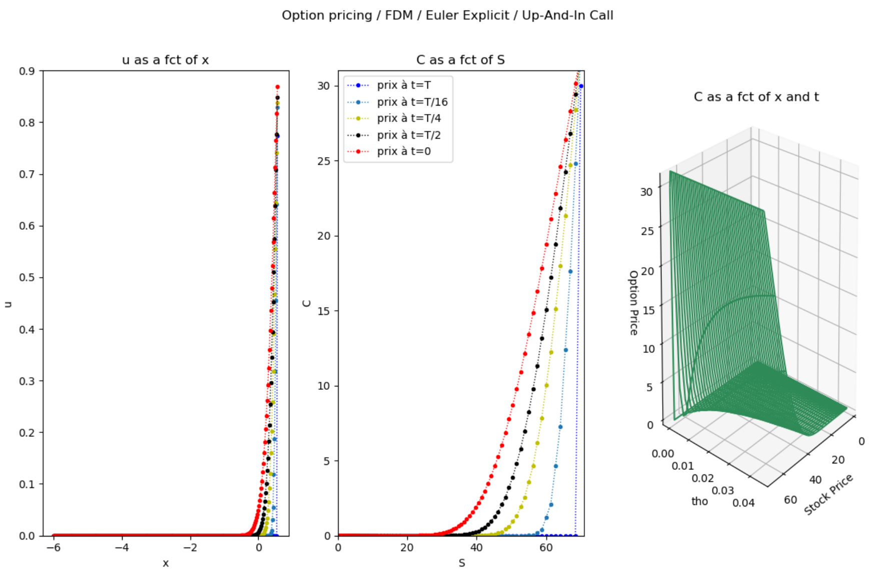

FIGURE 1.9 : FDM - Up and In Option Pricing using the Heat Eqt - Explicit

1.3.1 Comparison of convergences in space

We used there the same measure of error as for European Option and obtained the following result.

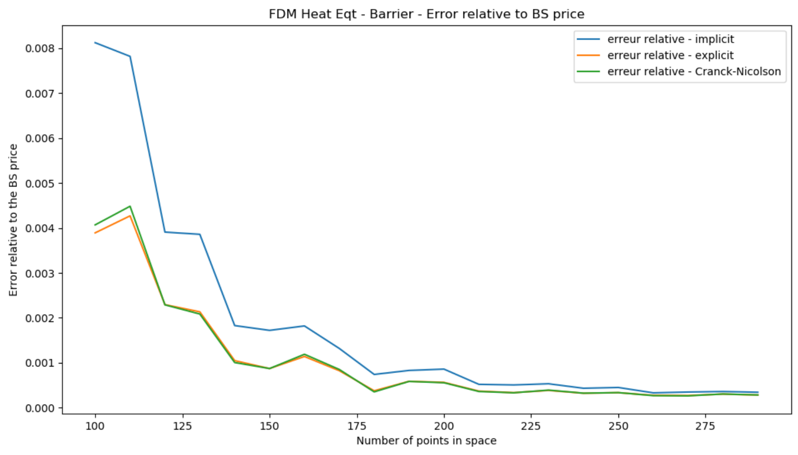

FIGURE 1.10 : FDM - Barrier Option - Relative Errors for the 3 schemes using the Heat Equation

1.3.2 Barrier Parity

We would like to check the property of parity for barrier options, which stands in this case that the sum of the Up-And-Out and the Up-And-In is a European option with the same parameters. This is what we obtain when computing the sum of our Finite Difference solutions for both barriers. Here is what we can obtain graphically with the Explicit Scheme:

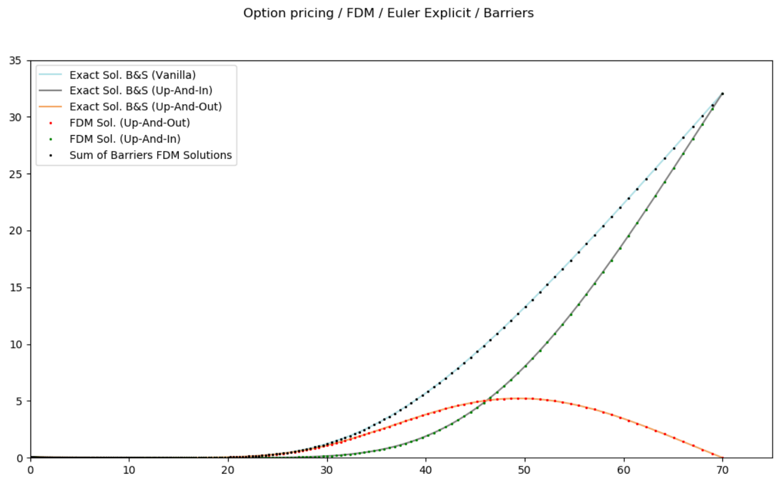

Figure 1.11: Sum of Barriers numerical solutions with the Explicit Scheme

2 Finite Element Method

For the finite element method, we will use the change of variable leading to the heat equation, as done before with the finite difference method. However, instead of two boundary conditions on the values of the lower and upper spatial coordinates, we use a "natural" condition and an "essential" condition. For a call option, the initial condition is still

$$u(x, \tau = 0) = max(e^{\frac{(k+1)}{2}x} - e^{\frac{(k-1)}{2}x}, 0)$$

and the essential conditions are

- $u(x_{min}, \tau) = 0$
- $\dfrac{\partial u}{\partial x}(x_{max}, \tau) = \dfrac{1}{2}((k+1)e^{x_{max}} - (k-1)e^{-k\tau})e^{\alpha x_{max} + \beta\tau}$

We will first use a spatial discretization using a constant mesh length $\Delta\tau$ and will then adapt it to our problem. For the time discretization, we will use the same kind of discretization as in the finite difference method.

2.1 Using Linear Lagrangian form functions

Let's first fix a time coordinate τ. On each interval $[x_j; x_{j+1}]$, we use the two following functions :

$$\begin{array}{ll} [-1,1] \to \mathbb{R} & [-1,1] \to \mathbb{R} \\ \xi \overset{N_{2j}}{\mapsto} \dfrac{1-\xi}{2} & \xi \overset{N_{2j+1}}{\mapsto} \dfrac{1+\xi}{2} \end{array}$$

Let N be the number of points in our spatial discretization, we then have $N-1$ intervals and thus $2N-2$ shape functions. Therefore, on the interval $[x_j; x_{j+1}]$,

$$u(x(\xi),\tau) = N_{2j}(\xi)q_{2j}(\tau) + N_{2j+1}(\xi)q_{2j+1}(\tau)$$

where ξ is such that $x = \frac{1}{2}(1-\xi)x_i + \frac{1}{2}(1+\xi)x_{j+1}$ (bijection between x and ξ, for any x in $[x_j, x_{j+1}]$.)

2.1.1 Variation formulation of the PDE

If we fix the time variable τ, we can write

$$a(u,v) = \int_{L_{min}}^{L_{max}} (\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial \tau})v(x)dx = 0$$

for any v in, say, $\mathscr{C}([L_{min}, L_{max}])$. This is the variation formulation of the equation. Here we consider the subspace of the piecewise linear function, whose basis functions are the $2N-2$ Lagrangian form functions defined above. v is then any linear combination of those shape functions. As the shape functions form a basis of the space of piecewise linear functions, this can be re-written in the form:

$$\forall k \in [0, 2N-3], \quad \int_{L_{min}}^{L_{max}} (\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial \tau})N_k(\xi)dx = \int_{x_j}^{x_{j+1}} (\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial \tau})N_k(\xi)dx = 0$$

Let's fix k in $[0, N-1]$, then let $N^T = \begin{pmatrix} N_{2k} \\ N_{2k+1} \end{pmatrix}$, $q = \begin{pmatrix} q_{2k} \\ q_{2k+1} \end{pmatrix}$, $X = \begin{pmatrix} x_k \\ x_{k+1} \end{pmatrix}$, then $u = N.q$ and $x = N.X$, with these notations, the variation formulation can be written, for each finite element:

$$\int_{x_j}^{x_{j+1}} N^T(\frac{\partial^2(N.q)}{\partial x^2} - \frac{\partial(N.q)}{\partial \tau})dx = 0$$

Let's do a change of variable to obtain a integral on ξ:

$$\int_{-1}^{1} N^T(\frac{\partial^2(N.q)}{\partial x^2} - \frac{\partial(N.q)}{\partial \tau})\frac{dx}{d\xi}d\xi = \int_{-1}^{1} N^T(\frac{\partial^2(N.q)}{\partial x^2} - \frac{\partial(N.q)}{\partial \tau})\frac{\Delta x}{2}d\xi = 0$$

And by integrating by part the first integral:

$$\frac{\Delta x}{2}[N^T\frac{\partial N}{\partial x}q]_{-1}^{1} - \frac{\Delta x}{2}\int_{-1}^{1} \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial x}q d\xi - \frac{\Delta x}{2}\int_{-1}^{1} N^T N.\frac{\partial q}{\partial \tau}d\xi = 0$$

Let's denote

$$F^e := \frac{\Delta x}{2}[N^T\frac{\partial N}{\partial x}q]_{-1}^{1} = \frac{\Delta x}{2}\begin{pmatrix} N_{2k}(1)\frac{\partial u_{j+1}}{\partial x} - N_{2k}(-1)\frac{\partial u_j}{\partial x} \\ N_{2k+1}(1)\frac{\partial u_{j+1}}{\partial x} - N_{2k+1}(-1)\frac{\partial u_j}{\partial x} \end{pmatrix} = \frac{\Delta x}{2}\begin{pmatrix} -\frac{\partial u_j}{\partial x} \\ \frac{\partial u_{j+1}}{\partial x} \end{pmatrix}$$

This is the vector containing the natutal boundary conditions evaluated at the extreme nodes of the element. Then if we denote $B := \frac{\partial N}{\partial x}$, $C^e := \frac{\Delta x}{2}(\int_{-1}^{1} N^T N d\xi)$ and $K^e = \frac{\Delta x}{2}(\int_{-1}^{1} B^T B d\xi)$, we have :

$$F^e = \frac{\Delta x}{2}(\int_{-1}^{1} B^T B d\xi)q + \frac{\Delta x}{2}(\int_{-1}^{1} N^T N d\xi)\frac{\partial q}{\partial x} = K^e q + C^e\frac{\partial q}{\partial x} \ (\triangle)$$

Moreover, it can be shown by simple computation that $C^e = \frac{\Delta x}{3}\begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}$ and $K^e = \frac{1}{\Delta x}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. Equation $(\triangle)$ is only true at the element scale, the system of equations for the entire finite element system is represented by:

$$Kq + C\frac{\partial q}{\partial x} = F \ (\diamondsuit)$$

where K, C and F are $(2N-2) \times (2N-2)$ matrices, built using all the 2×2 element matrices, as follows:

$$K = \begin{pmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & & & & 0 \end{pmatrix} + ... + \begin{pmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \cdots & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & 0 \\ \vdots & & \ddots & 0 & 1 & -1 \\ 0 & & & 0 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix}$$

Then we use a backward finite difference approximation for the first order derivative of the time variable: $\frac{\partial q^i}{\partial \tau} = \frac{q^i - q^{i-1}}{\Delta \tau}$ and obtain :

$$(\frac{1}{\Delta \tau}C + K)q^i = (\frac{1}{\Delta \tau}C)q^{i-1} + F$$

which can be solved to find q^i. We can then compute the coefficients of the different form functions at inception, and thus the different value of v at the nodes, noticing that $\forall j, u_j = q_{2j}$ and $u_{j+1} = q_{2j+1}$

2.1.2 Implementation

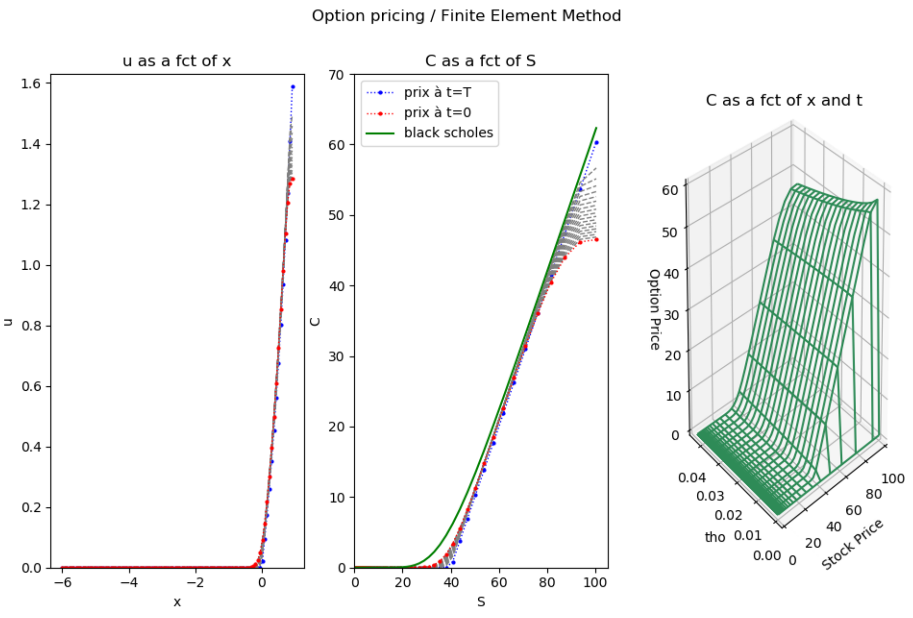

Figure 2.1: FEM - European Option - Lagrangian Linear Functions

2.2 Using Quadratic form functions

Let's now use quadratic form functions instead of linear functions. Here we define those functions for three space coordinates. For three points $x_{2j}, x_{2j+1}, x_{2j+2}$, we use the three following functions:

$$\begin{array}{ccc} [-1,1] \to \mathbb{R} & [-1,1] \to \mathbb{R} & [-1,1] \to \mathbb{R} \\ \xi \overset{N_{3j}}{\mapsto} \dfrac{-\xi(1-\xi)}{2} & \xi \overset{N_{3j}}{\mapsto} (1-\xi)(1+\xi) & \xi \overset{N_{3j+2}}{\mapsto} \dfrac{\xi(1+\xi)}{2} \end{array}$$

Let still use N as the number of points in our spatial discretization, we then have $3 \times \frac{N-3}{2} + 3$ shape functions (indexed from 0 to $3 \times \frac{N-3}{2} + 2$). The use of quadratic form functions also creates a new constraint on N : it has to be odd. Intuitively, if, as in Python, we index our points from 0 to $N-1$, the boundaries (index) of each three-points element have to be even as we start from 0. In particular, the boundary of the last element, $N-1$, has to be odd.

Then, on the interval $[x_{2j}; x_{2j+2}]$,

$$u(x(\xi),\tau) = N_{3j}(\xi)q_{3j}(\tau) + N_{3j+1}(\xi)q_{3j+1}(\tau) + N_{2j+2}(\xi)q_{3j+2}(\tau)$$

where ξ is such that $x = -\frac{1}{2}(1-\xi)\xi x_{2j} + (1-\xi)(1+\xi)x_{2j+1} + \frac{1}{2}(1+\xi)\xi x_{2j+2}$ (bijection between x and ξ, for any x in $[x_{2j}, x_{2j+2}]$.)

2.2.1 Varation Formulation of the PDE

We obtain the variation formulation in the same way as before. Howerver, the matrix formulation is similar but with different matrices. If we fix the time variable τ, we can write

$$a(u,v) = \int_{L_{min}}^{L_{max}} (\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial \tau})v(x)dx = 0$$

for any v in, say, $\mathscr{C}([L_{min}, L_{max}])$. This is the variation formulation of the equation. Here we consider the subspace of the piecewise linear function [///////?///////], whose basis functions are the $3\times\frac{N-3}{2}+3$ quadratic form functions defined above. v is then any linear combination of those shape functions. As the shape functions form a basis of the space of piecewise linear functions [///////?///////], this can be re-written in the form:

$$\forall k \in [0, \frac{3}{2}(N-3)+2], \quad \int_{L_{min}}^{L_{max}} (\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial \tau})N_k(\xi)dx = \int_{x_j}^{x_{j+1}} (\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial \tau})N_k(\xi)dx = 0$$

Let's fix k in $[0, N-1]$, then let $N^T = \begin{pmatrix} N_{3k} \\ N_{3k+1} \\ N_{3k+2} \end{pmatrix}$, $q = \begin{pmatrix} q_{3k} \\ q_{3k+1} \\ q_{3k+2} \end{pmatrix}$, $X = \begin{pmatrix} x_{2k} \\ x_{2k+1} \\ x_{2k+2} \end{pmatrix}$, then $u = N.q$ and $x = N.X$, with these notations, the variation formulation can be written, for each finite element:

$$\int_{x_j}^{x_{j+1}} N^T(\frac{\partial^2(N.q)}{\partial x^2} - \frac{\partial(N.q)}{\partial \tau})dx = 0$$

Let's do a change of variable to obtain a integral on ξ:

$$\int_{-1}^{1} N^T(\frac{\partial^2(N.q)}{\partial x^2} - \frac{\partial(N.q)}{\partial \tau})\frac{dx}{d\xi}d\xi = \int_{-1}^{1} N^T(\frac{\partial^2(N.q)}{\partial x^2} - \frac{\partial(N.q)}{\partial \tau})(\Delta x)d\xi = 0$$

And by integrating by part the first integral:

$$\Delta x[N^T \frac{\partial N}{\partial x} q]_{-1}^{1} - \Delta x \int_{-1}^{1} \frac{\partial N^T}{\partial x} \frac{\partial N}{\partial x} q d\xi - \Delta x \int_{-1}^{1} N^T N . \frac{\partial q}{\partial \tau} d\xi = 0$$

Let's denote

$$F^e := \Delta x[N^T \frac{\partial N}{\partial x} q]_{-1}^{1} = \Delta x \begin{pmatrix} N_{2k}(1)\frac{\partial u_{j+1}}{\partial x} - N_{2k}(-1)\frac{\partial u_j}{\partial x} \\ N_{2k+1}(1)\frac{\partial u_{j+1}}{\partial x} - N_{2k+1}(-1)\frac{\partial u_j}{\partial x} \end{pmatrix} = \Delta x \begin{pmatrix} -\frac{\partial u_j}{\partial x} \\ \frac{\partial u_{j+1}}{\partial x} \end{pmatrix}$$

This is the vector containing the natutal boundary conditions evaluated at the extreme nodes of the element. Then if we denote $B := \frac{\partial N}{\partial x}$, $C^e := \frac{\Delta x}{2}(\int_{-1}^{1} N^T N d\xi)$ and $K^e = \frac{\Delta x}{2}(\int_{-1}^{1} B^T B d\xi)$, we have :

$$F^e = \frac{\Delta x}{2}(\int_{-1}^{1} B^T B d\xi) q + \frac{\Delta x}{2}(\int_{-1}^{1} N^T N d\xi)\frac{\partial q}{\partial x} = K^e q + C^e \frac{\partial q}{\partial \tau} \ (\triangle)$$

Moreover, it can be shown by simple computation that $C^e = \frac{\Delta x}{3}\begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}$ and $K^e = \frac{1}{\Delta x}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. Equation $(\triangle)$ is only true at the element scale, the system of equations for the entire finite element system is represented by:

$$Kq + C\frac{\partial q}{\partial \tau} = F \ (\diamondsuit)$$

where K, C and F are $(2N-2) \times (2N-2)$ matrices, built using all the 2×2 element matrices, as follows:

$$K = \begin{pmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & & & & 0 \end{pmatrix} + \ldots + \begin{pmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \cdots & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & 0 \\ \vdots & & \ddots & 0 & 1 & -1 \\ 0 & & & 0 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix}$$

Then we use a backward finite difference approximation for the first order derivative of the time variable: $\frac{\partial q^i}{\partial \tau} = \frac{q^i - q^{i-1}}{\Delta \tau}$ and obtain :

$$(\frac{1}{\Delta \tau}C + K)q^i = (\frac{1}{\Delta \tau}C)q^{i-1} + F$$

which can be solved to find q^i. We can then compute the coefficients of the different form functions at inception, and thus the different value of v at the nodes, noticing that $\forall j, u_j = q_{2j}$ and $u_{j+1} = q_{2j+1}$

2.2.2 Implementation

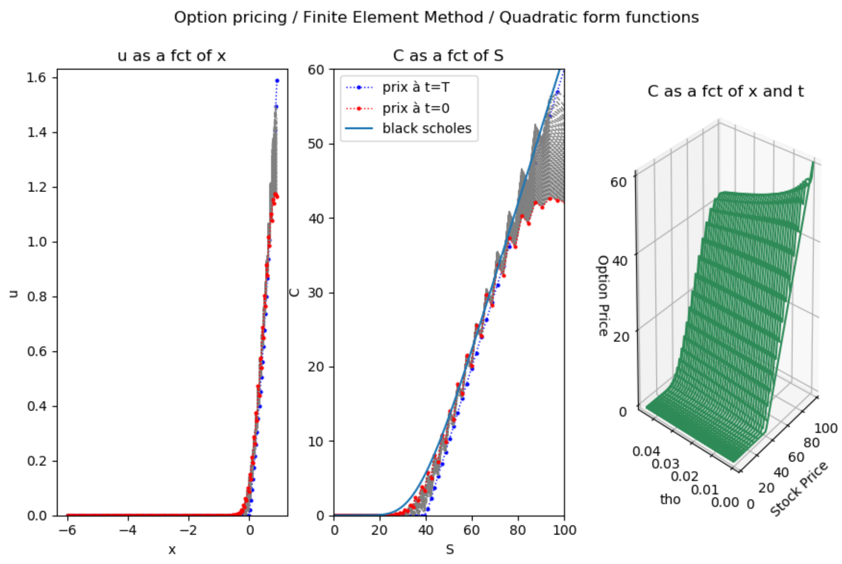

Figure 2.2: FEM - European Option - Quadratic Functions