



EDP - Projet

Pablo THOMASSIN, Santiago VAZART¹

¹ESILV-A3-S6-TD_S

Application du modèle des équations de la chaleur à la finance, l'équation de Black-Scholes.

Aujourd'hui de nombreux domaines de l'industries s'intéresse aux équations aux dérivés partiels : de la modélisation de flux d'eau dans des barrages à création d'IA via la recherche d'optimum. Néanmoins depuis l'émergence des marchés financier, marché des matières premières dans un premier temps, de nombreux mathématicien et physicien se sont intéressé à des modélisation pouvant décrire les comportements d'un marché, notamment sont évolution future dans l'objectif d'engendrer des bénéfices. Depuis 1973 un modèle mathématique semble dominer l'interprétation des marchés : Le Modèle de Black-Scholes-Merton décrivant le prix d'une option selon un processus stochastique en temps continue par opposition au modèle de Cox-Ross-Rubinstein traitant le problème en temps discret. Nous nous demanderons en quoi le modèle de Black-Scholes-Merton apprécie de manière satisfaisante l'évolution du prix d'une option vis-à-vis de l'évolution future de l'action. Pour cela nous réaliseront une introduction mathématique préliminaire permettant de poser le modèle. Puis nous démontrerons à la fois des solutions théoriques ainsi qu'analytique via une simulation python. Finalement nous élargiront le sujet en prenant en compte des critique émise par la communauté scientifique et notamment par M. J.P.Bouchaud.[4]

Table des matières

1	Modélisation mathématique	2
1.1	Vocabulaire et notations	2
1.2	Formulation de l'équation de Black-Scholes	2
1.3	Conditions aux limites	2
1.3.1	Condition aux limites pour l'achat d'une option (call)	2
1.3.2	Condition aux limites pour la vente d'une option (put)	3
1.4	Lien entre l'équation de la chaleur et l'équation de Black-Scholes	3
2	Analyse mathématique et numérique	4
2.1	Analyse mathématique	4
2.2	Méthode des différences finies et schéma	4
2.2.1	Création de la grille	4
3	Python	6
4	Pour aller plus loin : les critiques émises par l'éconophysique	9
5	Conclusion	9

1 Modélisation mathématique

1.1 Vocabulaire et notations

$$\left\{ \begin{array}{ll} V & \text{la valeur d'une option (ici européenne)} \\ \sigma & \text{la volatilité du prix de l'action} \\ E & \text{le prix de l'action fixé par l'option} \\ T & \text{le temps qui reste à l'option avant son échéance} \\ r & \text{le taux d'intérêt} \\ S & \text{le prix de l'option} \end{array} \right.$$

Option[2] : produit dérivé qui établit un contrat entre un acheteur et un vendeur. La différence entre une action et une option est la suivant : une action est un titre qui nous rends actionnaire de la société tandis qu'une option nous donne le droit d'acheter une action.

De plus : « L'acheteur de l'option obtient le droit, et non pas l'obligation, d'acheter (call) ou de vendre (put) un actif sous-jacent à un prix fixé à l'avance (strike), pendant un temps donné ou à une date fixée. Ce contrat peut se faire dans une optique de spéculation sur le prix futur de l'actif sous-jacent, ou d'assurance contre une évolution défavorable de ce prix. »

1.2 Formulation de l'équation de Black-Scholes

L'équation de Black-Scholes se formule de la façon suivante [4] :

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0 \quad (1.1)$$

On remarque directement que c'est une équation aux dérivés partiels d'ordre 2, de plus elle est de type parabolique (Comme l'équation de la chaleur, cf : 1.4 lien entre équation de la chaleur et équation de Black-Scholes)

Nous devons donc pour la définir correctement préciser une condition en temps (final ou initial), et des conditions aux limites en espaces.

1.3 Conditions aux limites

1.3.1 Condition aux limites pour l'achat d'une option (call)

On commence par poser C la fonction définissant l'évolution d'un call, c'est une fonction $C = C(S, t)$, soit du sous-jacent S et du temps (De la même manière $V = V(S, t)$).

Dans notre cas d'étude (option européenne), cette condition est finale, en effet le temps $t = T$ correspond à l'échéance de l'option pour un call de valeur (à t fixé donc) :

$$C(S, T) = \max(S - E, 0), \quad \forall S$$

Cas 1 : si $S < E$, il n'est pas intéressant d'exercer (donc d'acheter l'action) cela engendre une perte $E - S$

Cas 2 : si $S > E$ alors, on engendre un bénéfice $S - E$.

Maintenant dans le cas où étudie par rapport à S :

Cas 1 : $S = 0$, alors de manière évidente le bénéfice est nul et donc il n'y aucun intérêt une fois de plus d'exercer l'option on obtient ainsi : $C(0, t) = 0, \forall t$

Cas 2 : $S \rightarrow +\infty$: Alors il vient que l'on peut négliger le prix d'exercice E ainsi : $C(S, t) \xrightarrow{S \rightarrow \infty} +\infty$
En utilisant cette condition et l'équation de Black-Scholes on détermine le prix du call.

1.3.2 Condition aux limites pour la vente d'une option (put)

On a donc $P = P(S, t)$ avec une condition finale qui est le bénéfice (à $t = T$ fixé) :

$$P(S, T) = \max(S - E, 0), \quad \forall S$$

Ici on reprends le même raisonnement logique qu'en 1.3.1 sauf que le cas 1 désigne $S > E$ et le cas 2 $S < E$.

De plus si on étudie par rapport à S on peut distinguer deux cas :

Cas 1 : Si $S \rightarrow +\infty$ alors $P(S, t) \rightarrow 0$, et donc on n'exerce pas l'option

Cas 2 : Si $S = 0$ alors sachant qu'à $t = T$, $P(0, T) = E$ et en considérant le taux d'intérêt r constant, il viens en résolvant l'équation de Black-Scholes pour $S = 0$:

$$P(0, t) = Ee^{-r(T-t)} \quad \forall t$$

Ainsi en utilisant l'équation de Black-Scholes on détermine le prix d'un put

1.4 Lien entre l'équation de la chaleur et l'équation de Black-Scholes

Pour montrer ce lien on utilise l'équation de Black-Scholes avec par exemple les conditions aux limites et finale d'un call, nous partons de :

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

De plus, C vérifie $C(0, t) = 0$, $C(S, t) \xrightarrow{S \rightarrow \infty} S$ et $C(S, T) = \max(S - E, 0)$.

Par un changement de variable dans le but de revenir en temps croissant en supprimant les termes en S et S^2 on pose :

$$\begin{cases} x = \ln\left(\frac{S}{E}\right) & \iff S = Ee^x & \implies x \in \mathbb{R} \\ \tau = 2\sigma^2(T-t) & \iff t = T - \frac{\tau}{2\sigma^2} & \implies \tau \in [0, \frac{\sigma^2 T}{2}] \\ C = Ev(x, t) \end{cases} \quad (1.2)$$

On obtient ainsi :

$$\frac{\partial v}{\partial \tau} = \frac{\partial^2 v}{\partial x^2} + (k-1) \frac{\partial v}{\partial x} - kv \quad \text{avec } k = \frac{2r}{\sigma^2} \quad (1.3)$$

On obtient la condition initiale : $v(x, 0) = \max(e^x - 1, 0)$

En extrayant l'équation caractéristique de 1.3 $\alpha^2 + (k-1)\alpha - k$, on pose α tel que :

$$\begin{cases} \alpha = -\frac{k-1}{2} \\ \beta = \alpha^2 + (k+1)\alpha - k \end{cases} \quad (1.4)$$

Ainsi, il vient finalement :

$$v(x, t) = e^{\alpha x + \beta \tau} u(x, \tau) \quad (1.5)$$

Ce qui permet de poser :

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} \quad (1.6)$$

Ce qui correspond à l'équation de la chaleur. On précisera la condition initiale :

$$u(x, 0) = u_0(x) = \max(e^{\frac{k+1}{2}x} - e^{\frac{k-1}{2}x}, 0) \quad (1.7)$$

Nous pouvons donc comme vue dans le cours, étudier les solutions analytiques et numérique ainsi que de proposer une méthode au différence finie en partant de cette formulation. Nous reviendront à l'équation de Black-Scholes en utilisant les changements de variable inverse [4] [3].

2 Analyse mathématique et numérique

2.1 Analyse mathématique

La solution à l'équation 1.6 est donnée par :

$$u(x, t) = \frac{1}{2\sqrt{\pi\tau}} \int_{-\infty}^{+\infty} u_0(s) e^{-\frac{(x-s)^2}{4\tau}} ds \quad (2.1)$$

Ainsi en remplaçant u_0 par son expression déterminée précédemment on obtient pour un call (ou un put) :

$$\begin{aligned} d_1 &= \frac{\ln(\frac{S}{E}) + (r + \frac{\sigma^2}{2})(T - t)}{\sigma\sqrt{T - t}} \\ d_2 &= \frac{\ln(\frac{S}{E}) + (r - \frac{\sigma^2}{2})(T - t)}{\sigma\sqrt{T - t}} \\ N(d) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^d e^{-\frac{s^2}{2}} ds \end{aligned}$$

Avec pour :

$$\begin{aligned} \text{Call : } C(S, t) &= SN(d_1) - Ee^{-r(T-t)}N(d_2) \\ \text{Put : } P(S, t) &= Ee^{-r(T-t)}N(-d_2) - SN(-d_1) \end{aligned}$$

Pour démontrer l'existence et l'unicité à l'équation de Black-Scholes on peut ainsi le montrer pour l'équation de la chaleur car par changement de variable tout coïncidera. Pour l'existence cela se prouve comme dans le cours par la méthode de séparation des variables, nous avons d'ailleurs fournie une solution générale en début de chapitre.

Pour l'unicité nous aurions pu faire le choix d'utiliser une méthode d'énergie avec deux solution u_1 et u_2 tel que $\omega = u_1 - u_2$ un couplage et de réécrire le système différentiel pour l'équation de la chaleur avec ce changement de variable, en posant ensuite l'énergie sous forme intégrale puis en dérivant sous celle-ci (espace compact) on obtient ; pour tout t positif $e(t) \leq e(0) = 0$. Sachant que e est positif il vient $e = 0$ et surtout $w = 0$ ce qui démontre bien l'unicité.

2.2 Méthode des différences finies et schéma

2.2.1 Création de la grille

Dans une premier temps, nous discrétisons notre grille spatio-temporelle. Nous discrétisons la dimension temporelle en M intervalles, puis la dimension spatiale en N intervalles. Cependant, $x \in \mathbb{R}$. Il nous faut donc limiter la région de l'espace sur l'intervalle $[A\delta x, B\delta x]$ avec A, B tels que $u(A\delta x, m) \approx u(-\infty, m)$ et $u(B\delta x, m) \approx u(+\infty, m)$

On pose :

$$\begin{cases} x = n\delta x, & x \in [A\delta x, B\delta x] \\ \tau = m\delta\tau, & \tau \in [0, \frac{\sigma^2 T}{2}] \end{cases} \quad \text{avec} \quad \begin{cases} \delta x = \frac{B-A}{N}; & A < 0 < B \\ \delta\tau = \frac{\sigma^2 T}{2M} \end{cases} \quad (2.2)$$

Par la suite, on note $u_m^n = u(n\delta x, m\delta\tau)$

Nous souhaitons utiliser le schéma de Cranck-Nicholson :

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

Cette approximation provient de la formule de Taylor en $O(\delta\tau^2 + \delta x^2)$. Pour des raisons d'implémentation on écrit l'équation sous la forme :

$$u_n^{m+1} - \frac{1}{2}\alpha(u_{n-1}^{m+1} - 2u_n^{m+1} + u_{n+1}^{m+1}) = u_n^m + \frac{1}{2}\alpha(u_{n-1}^m - 2u_n^m + u_{n+1}^m) \quad \text{avec} \quad \alpha = \frac{\delta\tau}{\delta x^2} \quad (2.3)$$

On cherche alors à résoudre le système linéaire suivant :

$$HU^{m+1} = b^m \quad (2.4)$$

avec

$$H = \begin{pmatrix} 1+\alpha & -\frac{\alpha}{2} & 0 & \cdots & 0 \\ -\frac{\alpha}{2} & 1+\alpha & -\frac{\alpha}{2} & & 0 \\ 0 & -\frac{\alpha}{2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\frac{\alpha}{2} \\ 0 & 0 & \cdots & -\frac{\alpha}{2} & 1+\alpha \end{pmatrix} \quad \text{et} \quad b^m = \begin{pmatrix} 0 \\ f_{A+1}^m \\ \vdots \\ f_0^m \\ \vdots \\ f_{B-1}^m \\ 0 \end{pmatrix} + \frac{\alpha}{2} \begin{pmatrix} u_A^{m+1} \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ u_B^{m+1} \end{pmatrix}$$

avec $f_n^m = (1-\alpha)u_n^m + \frac{1}{2}\alpha(u_{n-1}^m + u_{n+1}^m)$. Nous pouvons donc calculer $U^{m+1} = H^{-1}b^m$. En effet, en initialisant b^m à l'instant $m = 0$, ainsi que les conditions aux limites à l'instant $m = 1$, nous pouvons déterminer u à l'instant $m + 1$. On a :

$$U^0 = u[:, 0] = \max \left((e^{\frac{k+1}{2}x} - e^{\frac{k-1}{2}x}), 0 \right) \quad (2.5)$$

De plus,

$$\begin{aligned} C(S, t) &\xrightarrow{S \rightarrow 0} 0 & C(S, t) &\xrightarrow{S \rightarrow \infty} S \\ \iff v(x, \tau) &\xrightarrow{S \rightarrow -\infty} 0 & v(x, \tau) &\xrightarrow{S \rightarrow \infty} \frac{S}{E} \\ \iff u(x, \tau) &\xrightarrow{S \rightarrow -\infty} 0 & u(x, \tau) &\xrightarrow{S \rightarrow \infty} \frac{S}{E} e^{-(\alpha x + \beta \tau)} \end{aligned}$$

On pose :

$$\begin{aligned} u_A^m &= u[0, m] = 0 \\ u_B^m &= u[-1, m] = (S/E) * \text{np.exp}(\alpha * x[-1] + \beta * \tau[m]) \end{aligned}$$

Finalement, nous revenons aux variables de base liées à l'équation de Black-Scholes :

$$S = Ee^x \quad (2.6)$$

$$t = T - \frac{\tau}{2\sigma^2} \quad (2.7)$$

$$C(S, t) = Ee^{\alpha x + \beta \tau} u(x, \tau) \quad (2.8)$$

3 Python

```

1 #Import de different module pour le fonctionnement du code
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.pyplot as plt
5 from mpl_toolkits.mplot3d import Axes3D
6
7 # Definition des variables Black-Scholes
8 sigma = 0.5 # volatilité du prix de l'action
9 E = 95 # prix d'exercice de l'option
10 T = 3/12 # temps avant echeance
11 r = 0.01 # taux d'interet
12 S = 100 # prix actuel de l'actif sous-jacent
13 N = 100 # resolution de la simulation
14
15
16 """
17     Differences finies
18 """
19
20 def black_scholes_matrix(S, T, r, E, sigma, N):
21
22     # Variables
23     s = np.linspace(0.01, S+S*0.1, N)
24     t = np.linspace(0.01, T+T*0.1, N)
25
26     # Changement des variables
27     k = 2*r / sigma**2
28     alpha1 = (1-k) / 2
29     beta = alpha1**2 + (k-1)*alpha1 - k
30     tau_max = T * sigma**2 / 2
31
32     # Espace dual
33     x = np.log(s/E)
34     tau = 2*sigma**2 * (T-t)
35
36     # Differentielles
37     dtau = tau_max/N
38     dx = (x[-1] - x[0])/N
39     alpha = dtau / dx**2
40
41     u = np.zeros((x.size, tau.size)) # definition de la matrice u
42
43     # Creation de la matrice C
44     C = np.diag(np.ones(N)*alpha + 1) + np.diag(np.ones(N-1)*(-alpha/2), 1) + np.diag(np.ones(N-1)*(-alpha/2), -1)
45
46     # Inversion de la matrice C
47     C_inv = np.linalg.inv(C)
48
49     # Initialisation
50     u[:,0] = np.maximum(np.exp((k+1)*x/2) - np.exp((k-1)*x/2), 0)
51
52     # Calcul de u(x, tau) avec un schema de crank Nicholson
53     for m in range(1, N):
54         u[0, m] = 0
55         u[-1, m] = (S/E) * np.exp(alpha*x[-1] + beta*tau[m])
56         b = np.zeros(N)
57         for n in range(1, N-1):
58             b[n] = (1-alpha)*u[n,m-1] + (1/2)*(u[n-1,m-1]+u[n+1,m-1])
59         u[:,m] = C_inv @ b.T
60
61     #Redefinition de varibale pour l'affichage graphique
62     s = E * np.exp(x)
63     t = T - tau / (2*sigma**2)
64     Call = E * u @ np.exp(-alpha * x[:, np.newaxis] - beta * tau)
65
66     #Definition de la grille d'affichage
67     S, t = np.meshgrid(np.log(S), t)
68

```

```

69 #Affichage en fonction du temps
70 fig = plt.figure(figsize=(10, 8))
71 ax = fig.add_subplot(111, projection='3d')
72 ax.plot_surface(t, s, np.log(E * np.exp(-alpha * x[:, np.newaxis] - beta * tau) @ u), cmap='
coolwarm')
73 ax.set_xlabel('temps a maturite')
74 ax.set_ylabel("prix de l'action")
75 ax.set_zlabel("prix du call")
76 ax.set_title('C en fonction de S et t')
77 plt.show()
78
79 return Call, s, t, alpha, beta
80
81 #Seconde fonction pour un affichage en fonction de la volatilitée
82 def black_scholes_value(S, T, r, E, sigma, N):
83     # Variables
84     s = np.linspace(0.01, S+S*0.1, N)
85     t = np.linspace(0.01, T+T*0.1, N)
86
87     # Changement des variables
88     k = 2*r / sigma**2
89     alpha1 = (1-k) / 2
90     beta = alpha1**2 + (k-1)*alpha1 - k
91     tau_max = T * sigma**2 / 2
92
93     # Espace dual
94     x = np.log(s/E)
95     tau = 2*sigma**2 * (T-t)
96
97     # Differentielles
98     dtau = tau_max/N
99     dx = (x[-1] - x[0])/N
100     alpha = dtau / dx**2
101
102     u = np.zeros((x.size, tau.size))
103
104     # Creation de la matrice C
105     C = np.diag(np.ones(N)*alpha + 1) + np.diag(np.ones(N-1)*(-alpha/2), 1) + np.diag(np.ones(N
-1)*(-alpha/2), -1)
106
107     # Inversion de la matrice C
108     C_inv = np.linalg.inv(C)
109
110     # Initialisation
111     u[:,0] = np.maximum(np.exp((k+1)*x/2) - np.exp((k-1)*x/2), 0)
112
113     # Calcul de u(x, tau)
114     for m in range(1, N):
115         u[0, m] = 0
116         u[-1, m] = (S/E) * np.exp(alpha*x[-1] + beta*tau[m])
117         b = np.zeros(N)
118         for n in range(1, N-1):
119             b[n] = (1-alpha)*u[n,m-1] + (1/2)*(u[n-1,m-1]+u[n+1,m-1])
120         u[:,m] = C_inv @ b.T
121
122     s = E * np.exp(x)
123     t = T - tau / (2*sigma**2)
124     Call = E * np.exp(-alpha * x[:, np.newaxis] - beta * tau) @ u
125
126     print("...")
127     print(Call[-1, -1])
128
129     return Call[-2, -2]
130
131 (call, s, t, alpha, beta) = black_scholes_matrix(S, T, r, E, sigma, N)
132
133
134 """ Affichage en fonction de la volatilité """
135
136 _sigma = np.linspace(0.0001, 1, N)
137 e = np.linspace(E-10, E+10, N)
138

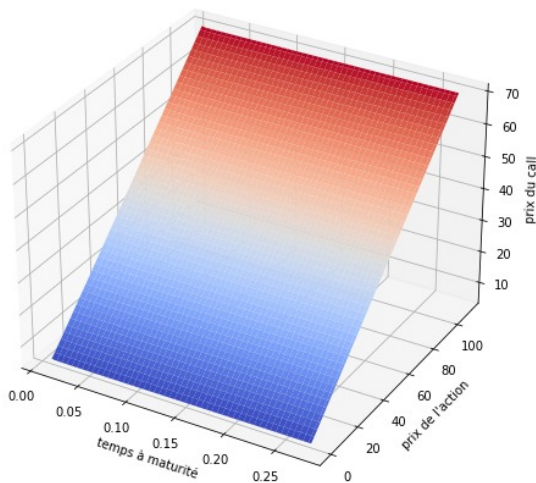
```

```

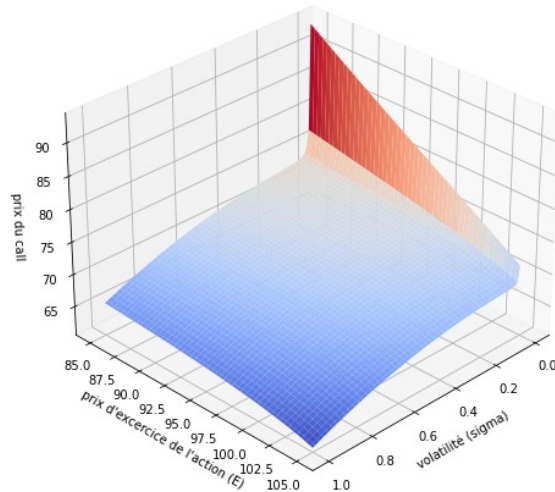
139 Z=[]
140 for sigma in _sigma:
141     Z_row = []
142     for E in e:
143         Z_row.append(black_scholes_value(S, T, r, E, sigma, N))
144     Z.append(Z_row)
145
146 X, Y = np.meshgrid(_sigma, e)
147 Z = np.array(Z).T
148
149 print(X)
150 print(Y)
151 print(Z)
152
153
154 fig = plt.figure(figsize=(10, 8))
155 ax = fig.add_subplot(111, projection='3d')
156 ax.plot_surface(X, Y, np.log(Z), cmap='coolwarm')
157 ax.set_xlabel('volatilite (sigma)')
158 ax.set_ylabel("prix d'exercice de l'action (E)")
159 ax.set_zlabel("prix du call")
160 ax.set_title('Prix du call en fonction de sigma et E')
161 ax.view_init(30, 45)
162 plt.show()

```

C en fonction de S et t



Prix du call en fonction de sigma et E



Dans ce graphique[1] nous pouvons observer la relation entre le prix de l'option, le temps restant jusqu'à l'échéance et la volatilité du sous jacent. De plus l'axe des x représente le temps restant jusqu'à l'échéance, l'axe des y représente le prix du sous jacent et l'axe des z représente le prix de l'option. Ainsi nous pouvons observer que plus notre prix est élevée plus la volatilité s'envole. De même plus le temps à échéance est grand plus le prix de l'option est élevée. Et cela est encore le cas au plus une action à un prix d'exercice (soit un prix initial grand). Tout cela est logique, car, un émetteur d'option n'est autre qu'un émetteur d'assurance mais pour l'achat ou la vente (future) d'une action. On effectue donc un report de risque quand on propose de vendre des options sur des actions. Ainsi plus la volatilité ou le temps d'échéance est grand plus le risque est grand et donc plus le prix de notre option doit être grand pour obtenir une stratégie de couverture à risque nulle.

De plus on remarque que lors de la vitesse de convergence correspond à celle théorique, nous avons donc pu aussi de par un écart à la solution théorique déterminer que notre solution est juste. Par ailleurs en comparant avec différentes solutions proposées par d'autres chercheurs nos graphiques correspondent. Avec certes une moins bonne définition et un jeu de données réduit.

4 Pour aller plus loin : les critiques émises par l'éconophysique

En effet, dans ses cours au collège de France M.Bouchaud émet de nombreuses critiques vis à vis de la modélisation fait par Black & Scholes, qui considère que le marché est efficient, cela signifie que le marché évolue selon une information non anticipé on parle de choc exogène. Pour autant certain estime que le choc serait endogène et donc lié aux transactions au sein du marché.

De plus le physicien remarque deux anomalies :

- La valeur de la prime ne dépend pas de la tendance anticipée (même en moyenne) du marché future. La question qui est soulevé est donc qu'en est il de la baisse future d'un marché ?

- Une stratégie de couverture impliquerai un risque nulle, or le marché est imprévisible selon le modèle probabiliste, néanmoins cet argument se justifie par l'hypothèse que le marché se comporte comme un mouvement Brownien (classiquement il s'agit d'une évolution par des micro pas réguliers) continue et donc d'une évolution continue sans saut. Or si on opte pour une queue de distribution épaisse le modèle se fragilise et on a donc un risque asymptotique et le fournisseur de l'option (soit nous au sein de ce sujet) prend plus de risque aussi.

5 Conclusion

Pour conclure, nous avons mis en place un modèle défini et proposé sa résolution à l'aide d'une équation connue : l'équation de la chaleur. Puis nous avons prit le temps d'analyser celle-ci ainsi que de l'implémenter par un code en python. Tout cela dans le but de pouvoir valoriser le prix d'une option. Nous pouvons ainsi, dans le cadre de la réglementation (si nous étions un établissement bancaires ou autre fond d'investissement ...) proposer un déplacement de risque pour nos usagers et leur permettre l'achat d'option pour des actions. Nous pourrions définir le prix de cette option pour couvrir tous les risques et ainsi engendrer un bénéfice en fournissant "une assurance" sur des actions.

Références

- [1] Documentation matplotlib.
- [2] Option (finance).
- [3] Équation de la chaleur.
- [4] Didier AUROUX. *Méthode numérique pour le pricing d'options*. POLYTECH'NICE-SOFIA, 2010.