

# Calibration par minimisation d'entropie relative d'après Avellaneda et al (1997)

Jacques Printems\*et Marouen Messaoud†

Janvier 2004

## 1 Introduction.

Il est bien connu que les travaux de Black & Scholes [BS73] constituèrent une avancée majeure dans le marché des produits dérivés. À partir d'une modélisation de l'évolution du sous-jacent par une équation différentielle stochastique (EDS),

$$\frac{dS_t}{S_t} = \sigma dW_t + \mu dt,$$

où  $\mu, \sigma$  sont des constantes positives et  $\{W_t\}$  un mouvement Brownien standard, ils montrèrent, grâce à une hypothèse d'absence d'opportunité d'arbitrage, que le prix  $C$  d'une option (de type européenne) basée sur ce sous-jacent vérifiait une équation aux dérivées partielles (EDP). Dans le cas d'un Call européen (option d'achat) d'échéance  $T$ , cette EDP s'écrit

$$C_t + \mu SC_s + \frac{1}{2}\sigma^2 S^2 C_{ss} - \mu C = 0, \quad s \in ]0, +\infty[, \quad t \in [0, T[, \quad (1.1)$$

avec la condition terminale

$$C(s, T) = (s - K)_+ = \max(s - K, 0), \quad (1.2)$$

où  $K > 0$  désigne le *prix d'exercice* de l'option.

Le modèle suscita immédiatement un vif intérêt dans les milieux financiers tant par le prix  $C(0, s) = \mathbb{E}(S_T - K)_+$  qu'il permet de donner à une option à  $t = 0$ , par exemple, que par la formule de *couverture* :

$$(S_T - K)_+ = \mathbb{E}(S_T - K)_+ + \int_0^T C_s(t, S_t) dS_t,$$

qui permet, connaissant  $C_s$ , et grâce à une richesse initiale (le prix de l'option), de gérer un portefeuille constitué du seul sous-jacent  $\{S_t\}$  afin de restituer à l'échéance  $T$  de l'option exactement la richesse nécessaire  $S_T - K - \mathbb{E}(S_T - K)_+$  à l'exercice éventuel de l'option. C'est le principe de partage du risque avec le marché. On parle de modèle *complet*.

Toujours dans le cadre du modèle de Black & Scholes, on peut à partir des prix d'options du marché retrouver la volatilité dite implicite du sous-jacent qui injectée dans le modèle redonne le prix observé. Malheureusement, il est bien connu que celle-ci varie d'un prix à l'autre : le modèle (1.1)-(1.2) n'est pas consistant avec le marché.

---

\*Laboratoire d'Analyse et de Mathématiques Appliquées & CMP12, CNRS UMR 8050, Université de Paris XII, 61, av. du Général de Gaulle, 94010 CRÉTÉIL

†Projet MathFi, INRIA-Rocquencourt, Bâtiment 12, Domaine de Voluceau, BP 105, 78153 LE CHESNAY

En effet, pour un même sous-jacent  $\{S_t\}$ , les acteurs du marché n'attribue pas le même risque à une option d'échéance courte qu'à une autre d'échéance plus longue. De même, le risque n'est pas apprécié identiquement en ce qui concerne les options «loin de la monnaie» ( $|S/K|$  grand) et «à la monnaie» ( $S \approx K$ ).

Dès lors, la préoccupation dans la recherche d'un modèle pour le sous-jacent est double : le modèle doit être le plus réaliste possible tout en permettant une couverture «exacte». Dans [HW87], les auteurs introduisirent un modèle à volatilité stochastique où la volatilité  $\{\sigma_t\}$  suit une EDS gouvernée par un autre mouvement Brownien. Ce modèle introduit une nouvelle source de bruit et par là de risque. Malheureusement ce risque n'est pas fongible car le processus  $\{\sigma_t\}$  n'est ni «achetable», ni «vendable».

Plus récemment, dans [Dup94], l'auteur propose de contraindre la volatilité à dépendre du seul sous-jacent  $\sigma_t = \sigma(t, S_t)$ , on parle de *volatilité locale*. Il montre que dans ce cadre, on peut se couvrir parfaitement en gérant un portefeuille constitué par le seul sous-jacent (le modèle est complet) tout en permettant de rendre compte des prix observés. C'est dans ce contexte alors que la problématique se déplace sur l'estimation de la nappe de volatilité  $(t, S) \mapsto \sigma(t, S_t)$  aux vues des prix d'options pratiqués : c'est le problème de la *calibration* (dans un modèle à volatilité locale). De cette façon, estime-t-on, le modèle incorporera les anticipations des acteurs du marché (les prix donnés) et donnera une image de la volatilité future. Il permettra également, au moins pour une courte période, de donner de nouveaux prix compatibles aux autres déjà fixés. Enfin, il permettra une couverture parfaite pour les prix donnés.

Signalons que c'est un problème inverse typiquement mal posé puisqu'il est bien connu que plusieurs modèles de volatilités peuvent donner des prix identiques (par exemple : une volatilité constante  $\sigma$  et une volatilité déterministe  $t \mapsto \sigma_t$  telle que  $\sigma^2 T = \int_0^T \sigma_t^2 dt$ ). On utilise alors généralement les techniques de régularisation de Tikhonov qui permettent de prendre en compte les «oscillations» de la volatilité dans le calibrage des données.

## 2 Entropie relative de deux mesures sur un espace de trajectoires. Cadre discret.

On s'intéresse ici à la technique de calibration de la nappe de volatilité locale introduite dans [AFHS97]. On fait un choix de diffusion :

$$\frac{dS_t}{S_t} = \sigma_t dZ_t + \mu dt \quad (2.1)$$

qui représente l'évolution du sous-jacent. On veut trouver au sein de cette famille de processus indexée par  $(\sigma_t)$  celui qui donne au mieux les prix de marché. Cette dernière contrainte se traduit par l'équation suivante :

$$\mathbb{E}^\sigma[e^{-rT_i} G_i(S_{T_i})] = C_i \quad (2.2)$$

(  $G_i(\cdot)$  et  $C_i$  sont le payoff et le prix des l'options).

Dans ce qui suit la diffusion dite *a priori* sera définie par une volatilité constante  $\sigma_0$  dans (2.1). Le choix de la probabilité *a priori* permet de prendre en compte le «consensus» autour de l'évolution future du marché. Dans [ABF<sup>+</sup>01], les auteurs suggèrent qu'à partir d'une simulation de  $n$  trajectoires  $S_1, \dots, S_n$  i.i.d. de (2.1) avec  $\sigma_t = \sigma_0$  on affecte des poids  $p_i$  aux trajectoires le plus proche possible de la loi uniforme  $1/n$  (ce qui revient à minimiser l'entropie statistique  $\sum_{i=1}^n p_i \ln(p_i)$ ) en faisant en sorte que les prix calculés avec cette nouvelle mesure soient les prix observés (Voir également [JN01] pour un résultat d'existence d'une telle mesure).

Dans [AFHS97], le critère choisi dans le cadre Markovien continue (2.1) par les auteurs est basé sur la distance Kullback-Leibler. L'idée est de prendre une diffusion *a priori* et de la modifier pour assurer la contrainte tout en contrôlant son éloignement (dans le sens de la distance choisie) par rapport à  $\sigma_0$  ( $\mu = r - q$  et  $\sigma_t$  est adaptée à la filtration brownienne).

Soit  $Q, P$  deux probabilités sur les espaces de trajectoires dans le cas continue de (2.1). Formellement, l'entropie relative de  $Q$  par rapport à  $P$  est définie de la manière suivante :

$$\varepsilon(Q, P) = \int_{\Omega} \ln\left(\frac{dQ}{dP}\right) dQ.$$

Malheureusement, dès que les processus de volatilités diffèrent dans (2.1), les deux mesures sont étrangères l'une à l'autre et l'intégrale précédente diverge. C'est pourquoi, on se place dans un cadre discret Markovien via l'approximation en loi de (2.1) par un arbre trinomial. Soit  $N$  un entier, on pose  $\Delta = T/N$ . La diffusion  $\{S_t\}$  est approchée en loi à la date  $t_n = nT/N$  par  $S_n$  définie de façon récursive par

$$S_{n+1} = S_n H_{n+1}, \quad n = 0, \dots, N,$$

où  $\{H_n\}_n$  est une suite i.i.d. de variables aléatoires telle que

$$H_{n+1} = \begin{cases} e^{\bar{\sigma}\sqrt{\Delta}}, & \text{avec la probabilité } P_U, \\ 1, & \text{avec la probabilité } P_M, \\ e^{-\bar{\sigma}\sqrt{\Delta}}, & \text{avec la probabilité } P_D, \end{cases}$$

où les probabilités sont paramétrées comme

$$\begin{aligned} P_U &= \frac{p}{2} \left(1 - \frac{\bar{\sigma}\sqrt{\Delta}}{2}\right) + \frac{\mu\sqrt{\Delta}}{2\bar{\sigma}}, \\ P_M &= 1 - p, \\ P_D &= \frac{p}{2} \left(1 + \frac{\bar{\sigma}\sqrt{\Delta}}{2}\right) - \frac{\mu\sqrt{\Delta}}{2\bar{\sigma}}, \end{aligned}$$

où  $\bar{\sigma} \geq \sigma_0$  est constante et où  $0 \leq p \leq 1$ . Par la suite,  $\bar{\sigma}$  restera fixe et  $p$  sera amené à varier entre 0 et 1.

Sur un pas de temps l'entropie relative d'une mesure  $Q$  sur les trajectoires par rapport à une mesure  $P$  *a priori* s'écrit

$$p \ln\left(\frac{p}{p_0}\right) + (1-p) \ln\left(\frac{1-p}{1-p_0}\right) + O(\Delta). \quad (2.3)$$

Par ailleurs, pour  $\Delta$  assez petit, le terme  $p\bar{\sigma}^2$  approche la variance de  $\ln(S_{t_n})$  rendant consistante l'approximation en loi des solutions de (2.1) avec  $0 < \sigma_{t_n} < \bar{\sigma}$ . Ainsi, si  $P$  désigne toujours la probabilité sur les trajectoires associée à la distribution *a priori*, elle sera caractérisée dans le cadre discret par le paramètre

$$p_0 = \frac{\sigma_0^2}{\bar{\sigma}^2}. \quad (2.4)$$

Quant à la volatilité associé à la mesure  $Q$ , elle varie sur l'arbre en fonction du paramètre  $p = p(t_n, x)$  comme

$$p(t_n) = \frac{\sigma_{t_n}^2}{\bar{\sigma}^2}. \quad (2.5)$$

L'entropie relative sur un pas de temps s'écrit au premier ordre en  $\Delta$  et compte tenu de (2.3)–(2.5) comme une fonction de  $\sigma^2$  :

$$\eta(\sigma^2) = \frac{\sigma^2}{\bar{\sigma}^2} \ln\left(\frac{\sigma^2}{\sigma_0^2}\right) + \left(1 - \frac{\sigma^2}{\bar{\sigma}^2}\right) \ln\left(\frac{\bar{\sigma}^2 - \sigma^2}{\bar{\sigma}^2 - \sigma_0^2}\right). \quad (2.6)$$

Pour  $N$  pas de temps, on dispose alors de  $3^N$  trajectoires et l'entropie relative doit être comprise en terme d'entropie statistique d'un système comportant  $N$  «micros-états» représentant à chaque intervalle de temps les changements de direction dans l'arbre. À ce titre, elle est proportionnelle à  $N$ .

### 3 Problème posé et algorithme.

#### 3.1 Structure lagrangienne.

Revenons formellement dans le cadre continu afin de poser le problème. À un facteur  $1/T$  près, il revient à minimiser

$$\min_{\sigma} \mathbb{E}^{\sigma} \int_0^T \eta(\sigma_s^2) ds \quad (3.1)$$

$$t.q \mathbb{E}^{\sigma} [e^{-rT_i} G_i(S_{T_i})] = C_i \quad i = 1..M; \quad (3.2)$$

Le problème dual s'écrit

$$\min_{\lambda_1 \dots \lambda_M \in \mathbb{R}^M} V(S_0, 0, \lambda_1, \dots, \lambda_M), \quad (3.3)$$

où on a posé

$$V(S_0, 0, \lambda_1, \dots, \lambda_M) = \sup_{\sigma} \mathbb{E}^{\sigma} \left[ - \int_0^T \eta(\sigma_s^2) ds + \sum_{i=1}^M \lambda_i e^{-rT_i} G_i(S_{T_i}) \right] - \sum_{i=1}^M \lambda_i C_i.$$

Dans [AFHS97] (Prop. 2, Section 3), les auteurs montrent que  $V$  satisfait une équation d'Hamilton-Jacobi-Bellman qui s'écrit de la manière suivante :

$$V_t + e^{rt} \Phi \left( \frac{e^{-rt}}{2} S^2 V_{SS} \right) + \mu S V_S - rV = - \sum_{t < T_i} \lambda_i \left( G_i(S_{T_i}) - e^{rT_i} C_i \right) \delta(t - T_i) \quad (3.4)$$

avec  $\Phi$  est la transformée de Legendre de  $\eta$  :

$$\Phi(x) = \sup_{\sigma_{min} \leq \sigma \leq \sigma_{max}} (\sigma^2 x - \eta(\sigma)) \quad (3.5)$$

#### 3.2 Algorithme.

Le calcul de  $V$  s'inspire des similitudes entre (1.1) et (3.4) en utilisant l'arbre trinomial décrit à la section précédente. On rappelle que les distances sur l'arbre sont fixes (déterminées par  $\bar{\sigma}$  et  $\Delta$ ) mais que les probabilités de transition changent d'un nœud à l'autre. Le calcul de  $V(S_0, 0, \dots)$  s'effectue d'une manière rétrograde en partant des termes sources de la date terminale et en ajoutant les termes sources aux dates où on dispose de prix d'option.

À partir d'une probabilité *a priori*, les transitions de probabilités vont être calculées à chaque étape de la procédure d'optimisation (décrite plus bas) comme suit. On détermine à partir du nouveau vecteur  $\lambda = (\lambda_1, \dots, \lambda_M)$  la fonction  $V$  grâce à l'arbre, puis  $\Gamma = 1/2s^2 V_{ss}$  comme

$$\Gamma_n^j = \frac{1}{\bar{\sigma}^2 \Delta} \left[ \left( 1 - \frac{\bar{\sigma} \sqrt{\Delta}}{2} \right) V_n^{j+1} - 2V_n^j + \left( 1 + \frac{\bar{\sigma} \sqrt{\Delta}}{2} \right) V_n^{j-1} \right].$$

Enfin, le paramètre  $p_n^j$  est déterminé par l'analogie de (2.5) dans le cas discret :

$$p_n^j = \frac{(\sigma_n^j)^2}{\bar{\sigma}^2}, \quad (3.6)$$

où la volatilité  $\sigma_n^j$  est déterminée grâce à  $\Phi$  et  $\Gamma_n^j$  :

$$\sigma_n^j = \frac{e^{rt_n} \Phi(e^{-rt_n} \Gamma_n^j)}{\Gamma_n^j}. \quad (3.7)$$

La recherche du minimum dans (3.3) s'effectue avec l'algorithme de quasiNewton BFGS. Pour cela on a besoin du gradient par rapport à  $\lambda$  de  $V$ . On note  $V_i = \frac{\partial V}{\partial \lambda_i}$ . Le calcul de ce gradient découle de l'équation d'HJB. En effet, formellement, une simple dérivation par rapport à  $\lambda$  donne :

$$V_{i,t} + \frac{1}{2} \Phi' \left( \frac{e^{-rt}}{2} S^2 V_{ss} \right) S^2 V_{i,ss} + \mu S V_{i,s} - r V_i = - \left( G_i(S_{T_i}) - e^{rT_i} C_i \right) \delta(t - T_i) \quad (3.8)$$

Le calcul de ce gradient s'effectue aussi de manière rétrograde sur l'arbre trinomial en adaptant toutefois (3.6)–(3.7) à la nouvelle équation :

$$p_n^j = \frac{1}{\sigma^2} \Phi' \left( e^{-rt_n} \Gamma_n^j \right).$$

### 3.3 Quelques heuristiques

Dans [AFHS97], les auteurs utilisent la fonction  $\eta(\sigma) = \frac{1}{2}(\sigma^2 - \sigma_0^2)^2$ . Il est apparu numériquement essentiel de disposer d'un degré de liberté  $\alpha$  sur  $\eta$  tel que  $\eta(\sigma) = \alpha(\sigma^2 - \sigma_0^2)^2$ . Faire varier  $\alpha$  revient à élargir la fenêtre (la bande passante) pour les valeurs de  $\Gamma = \frac{e^{-rt}}{2} S^2 V_{ss}$  comme le montre la figure 1. Cela permet numériquement de régulariser la surface de volatilité locale. Le cas extrême est de prendre  $\alpha = 0$  dans ce cas on retrouve la situation du modèle UVM (voir [ALP95]) où la volatilité ne peut prendre que les deux valeurs  $\sigma_{min}$  ou bien  $\sigma_{max}$ . En effet

$$\alpha = 0 \implies \Phi(\Gamma) = \sup_{\sigma_{min} \leq \sigma \leq \sigma_{max}} (\sigma^2 \Gamma)$$

donc

$$\begin{cases} si \Gamma > 0, & \sigma = \sigma_{max}; \\ si \Gamma < 0, & \sigma = \sigma_{min}. \end{cases}$$

Comme les tests numériques l'ont montré, la valeur de  $\alpha$  conditionne la rapidité de la convergence. En effet, on a représenté sur la figure 2, la norme du gradient de  $V$  en fonction des itérations du Quasi-Newton pour deux valeurs de  $\alpha$  et de  $\sigma_0$  ( $\alpha \in \{1, 100\}$  et  $\sigma_0 \in \{0.09, 0.19\}$ , pour une moyenne des volatilités implicites de 15%). On observe que la convergence est plus rapide pour  $\alpha = 100$  indépendamment de la valeur de  $\sigma_0$ . Par contre le choix de  $\alpha$  semble dépendre du jeu de données initial (cf. figure 3).

## 4 Documentation numérique

### 4.1 programme calib\_Avellaneda

Le programme calib\_Avellaneda calcule la volatilité locale  $\sigma(S, t)$  sur un arbre trinomial à partir des prix d'options,  $S_0, r, q$ , des aprioris  $\sigma_0, \sigma_{min}, \sigma_{max}$  et  $\bar{\sigma}$ . Cette calibration revient à minimiser (3.1) sous la contrainte (3.2). Pour cela il faut trouver le minimum en  $\lambda$  de  $V(S_0, 0, \lambda_1, \dots, \lambda_M)$  comme dans (3.3). Le programme utilise une méthode de QuasiNewton [Bar02] pour trouver ce minimum. Cette dernière routine nécessite le calcul de la fonction objective à minimiser dans ce cas  $V(S_0, 0, \lambda_1, \dots, \lambda_M)$  et de son gradient  $V_i = \frac{\partial V}{\partial \lambda_i}$ ,  $i = 1..M$ . Les équations (3.4) et (3.8) de ces deux dernières fonctions peuvent être résolues d'une manière rétrograde sur l'arbre trinomial en choisissant la bonne volatilité en chaque nœud. Dans l'équation (3.4) il faut choisir

$$\sigma(s, t) = \frac{\Phi(\frac{e^{-rt}}{2} S^2 V_{ss})}{e^{-rt} S^2 V_{ss}(s, t)}$$

Dans l'équation (3.8) il faut prendre

$$\sigma(s, t) = \Phi'(\frac{e^{-rt}}{2}s^2V_{ss}(s, t))$$

Dans les deux cas  $V_{ss}$  est donné sur chaque nœud de l'arbre par l'approximation suivante :

$$\frac{1}{2}s^2V_{ss}(s, t) = \frac{1}{\bar{\sigma}^2 dt}[(1 - \frac{\bar{\sigma}\sqrt{dt}}{2})V(s_u, t + dt) + (1 + \frac{\bar{\sigma}\sqrt{dt}}{2})V(s_d, t + dt) - 2V(s, t + dt)]$$

avec  $s_u > s > s_d$  sont les trois valeurs possibles que prend la diffusion sur l'arbre à partir du nœud  $(s, t)$  à la date  $t + dt$ . Le programme se déroule de la manière suivante :

1. chargement des prix des options et des différents paramètres;
2. creation de l'arbre trinomial;
3.  $\lambda^k = \lambda^0$ ;
4. calcul de  $V(S_0, 0, \lambda_1^k, \dots, \lambda_M^k)$ ;
5. calcul de  $V_i(S_0, 0, \lambda_1^k, \dots, \lambda_M^k)$ ;
6. calcul de  $\lambda^{k+1}$  dans la routine QuasiNewton;
7. test d'arrêt;
8. sinon :  $k = k + 1$  et retour à (4).

Le fichier d'entrée pour ce programme s'appelle "calib\_Avellaneda.in". Il a la forme suivante :

```
#####
##### VARIABLES S_0, r, q, N, sigma_0, sigma_min ,#####
##### sigma_max AND sigma_bar #####
#####
# S_0 : price of the underlying asset at t=t_0
100
# r : risk-free rate
0.05
# q : dividend (continuously compounded)
0
# N : number of space steps of the grid
100
# sigma_0 : --> prior
0.21
#sigma_min : --> least prior bound
0.10
# sigma_max : --> greatestprior bound
0.46
# sigma_bar : --> a default value must be greaterthen sigma_max
0.48
#####
##### OPTIMIZATION VARIABLES gradtol, steptol, verbosity, #####
##### saveSuccessiveXinFile,maxCounter and lambda #####
#####
# gradtol : tolerance on the relative gradient
0.00001
# steptol : tolerance on the relative change of x
0.00001
# verbosity : level of printed information (0 --> 3)
1
# saveSuccessiveXinFile : save successive x0 in the file data.out (0 or 1)
0
# maxCounter : maximum number of iterations
```

```

100
# lambda : Initial default value of lagrange parametre
0
# alpha : Tune this parametre when the program do not converge
1
#####
##### FILES IN/OUT #####
#####
# name_in_data : name of the file containing the data prices
VolStoPut.data
# name_out_Vol_Locale:name of the output file containing the Local Volatility
VolStoLoc.out
#####

```

N représente le nombre d'intervalle sur  $[0..T]$  :  $T = N * dt$ .

gradtol, stepTol, verbosity, saveSuccessiveXinFile, maxCounter et lambda sont les paramètres d'optimisation.

name\_in\_data est le nom du fichier contenant les prix des options. Ce fichier doit avoir la forme suivante :

```

80  0.5 0.3585  P
85  0.5 0.73546 P
90  0.5 1.4163  P
95  0.5 2.56686 P
100 0.5 4.37005 P
105 0.5 6.93777 P
110 0.5 10.2185 P
115 0.5 14.0489 P
120 0.5 18.2539 P
125 0.5 22.7012 P
80  1   1.24677 P
85  1   1.94357 P
90  1   2.93407 P
95  1   4.297   P
100 1   6.11092 P
105 1   8.42559 P
110 1   11.2437 P
115 1   14.5023 P
120 1   18.1192 P
125 1   22.0116 P

```

La première colonne = strike. La deuxième colonne = maturité. La troisième colonne = prix. Quatrième colonne = type de l'option (P pour Put, C pour Call). On peut inclure des 'puts' et des 'calls' dans n'importe quel ordre de maturité.

En sortie le programme écrit sur le terminal pour chaque option le strike, la maturité, le prix d'entrée, le prix calculé sur l'arbre calibrée, l'erreur relative en prix et la volatilité implicite correspondant au prix de la calibration.

```

K   T   Prix  PrixAvella  ErrRelPrix  VolImpAvella
80.000000 0.500000 0.358500 0.358572 0.000200 0.225057
85.000000 0.500000 0.735460 0.735271 0.000257 0.215992
90.000000 0.500000 1.416300 1.417292 0.000701 0.208040
95.000000 0.500000 2.566860 2.566917 0.000022 0.201694
100.000000 0.500000 4.370050 4.370505 0.000104 0.198201
105.000000 0.500000 6.937770 6.937133 0.000092 0.198145

```

```

110.000000 0.500000 10.218500 10.217814 0.000067 0.201057
115.000000 0.500000 14.048900 14.048730 0.000012 0.205845
120.000000 0.500000 18.253900 18.251965 0.000106 0.211337
125.000000 0.500000 22.701200 22.700776 0.000019 0.217218
80.000000 1.000000 1.246770 1.245136 0.001311 0.235509
85.000000 1.000000 1.943570 1.943707 0.000070 0.228471
90.000000 1.000000 2.934070 2.932568 0.000512 0.222245
95.000000 1.000000 4.297000 4.298210 0.000282 0.217450
100.000000 1.000000 6.110920 6.111125 0.000034 0.214302
105.000000 1.000000 8.425590 8.428096 0.000297 0.213305
110.000000 1.000000 11.243700 11.239824 0.000345 0.214245
115.000000 1.000000 14.502300 14.503043 0.000051 0.217056
120.000000 1.000000 18.119200 18.120107 0.000050 0.220855
125.000000 1.000000 22.011600 22.014958 0.000153 0.225366

```

Il crée aussi le fichier "name\_out\_Vol\_Locale" (VolStoLoc.out dans notre cas) où il stocke la volatilité locale sur l'arbre résultant de la calibration dans l'ordre " $s, t, \sigma(s, t)$ ". Ce dernier fichier est utile pour visualiser la volatilité locale. Il stocke aussi dans un fichier "arbre\_Avellaneda.out" les probabilités de transitions sur l'arbre. Il ne faut jamais modifier ou effacer ce fichier sinon le programme de 'pricing' risque ne pas fonctionner correctement.

Le dernier fichier créé s'intitule "pricer\_Avellaneda.in". Il est généré automatiquement pour sauvegarder les paramètres utilisés dans la calibration et nécessaires au 'pricing'. Ce fichier sera l'entrée pour le programme "pricer\_Avellaneda".

FICHIER : pricer\_Avellaneda.in

```

#####
#####Used Parameters To calibrate the Tree #####
#####S_0= 100.000 r= 0.050 dividende q = 0.000 N=100 #####
#####prior sigma_0=0.215 sigma_min=0.100 sigma_max=0.460 sigmabar=0.480 ###
#####
##### VARIABLES K, T, AND OPTIONTYPE #####
#####
# K : Strike of the option to price
125
# T : Maturity of the option to price must be less than the
# terminal date on the calibrated trinomial Tree =1.000
1
# optionType : type of the option (C for call, P for put)
P
#####
#####file containing the option to price#####
#####the data must be stored in this order #####
##### K T flag(P as put,Cas Call) #####
VolStoP.data
#####output file containing the price of these options#####
VolStoP.out
#####

```

## 4.2 programme price\_Avellaneda

Ce programme permet de 'pricer' à partir du dernier arbre trinomial calibré soit une seule option dont les paramètres sont édités dans le fichier

"pricer\_Avellaneda.in", soit une grille d'option dont les paramètres sont stockés dans un fichier (dans notre exemple c'est le fichier VolStoP.data) et le résultat sera écrit dans un autre fichier



dont le nom a été précisé par l'utilisateur dans le fichier "pricer\_Avellaneda.in". Des paramètres par défaut sont écrits dans le fichier d'entrée. C'est à l'utilisateur de les modifier ou de les effacer. Le nom du fichier contenant la grille des options est vide, même chose pour le nom du fichier de sortie contenant les résultats. S'il y a les paramètres d'une option le résultat est affiché sur le terminal.

Format du fichier VolStoP.data " Strike Maturité Type d'option(P,C) ".

```
80  0.5 P
85  0.5 P
90  0.5 P
95  0.5 P
100 0.5 P
105 0.5 P
110 0.5 P
115 0.5 P
120 0.5 P
125 0.5 P
80  1   P
85  1   P
90  1   P
95  1   P
100 1   P
105 1   P
110 1   P
115 1   P
120 1   P
125 1   P
```

Format du fichier de sortie VolStoP.out.

Prices Computed with the last output of calib\_Avellaneda  
(file arbre\_Avellaneda.out) from the data file VolStoPut.data

```
K T Price OptionType
80.000000 0.500000 0.358572 P
85.000000 0.500000 0.735271 P
90.000000 0.500000 1.417293 P
95.000000 0.500000 2.566917 P
100.000000 0.500000 4.370506 P
105.000000 0.500000 6.937134 P
110.000000 0.500000 10.217814 P
115.000000 0.500000 14.048730 P
120.000000 0.500000 18.251965 P
125.000000 0.500000 22.700776 P
80.000000 1.000000 1.245136 P
85.000000 1.000000 1.943707 P
90.000000 1.000000 2.932568 P
95.000000 1.000000 4.298211 P
100.000000 1.000000 6.111126 P
105.000000 1.000000 8.428097 P
110.000000 1.000000 11.239824 P
115.000000 1.000000 14.503044 P
120.000000 1.000000 18.120108 P
125.000000 1.000000 22.014958 P
```

## Références

- [ABF<sup>+</sup>01] M. Avellaneda, R. Buff, C. Friedman, N. Grandchamp, L. Kruk, and J. Newman. Weighted monte-carlo : A new technique for calibrating asset-pricing models. *Int. J. Theor. and Appl. Finance*, 4(1) :91–119, 2001.
- [AFHS97] M. Avellaneda, C. Friedman, R. Holmes, and D. Samperi. Calibrating volatility surfaces via relative-entropy minimization. *Applied Mathematical Finance*, 4(1), March 1997.
- [ALP95] M. Avellaneda, A. Levy, and A. Paras. Pricing and hedging derivatives securities in markets with uncertain volatilities. *Applied Mathematical Finance*, 2 :73–88, 1995.
- [Bar02] V. Barette. Un outil en C d’optimisation différentiable sans contraintes. Rapport de stage, Inria, Rocquencourt, France, 2002.
- [BS73] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. of Political Economy*, 81 :635–654, 1973.
- [Dup94] B. Dupire. Pricing with a smile. *Risk magazine*, 7 :18–20, 1994.
- [HW87] J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *Journal of Finance*, 3 :281–300, 1987.
- [JN01] B. Jourdain and L. Nguyen. Minimisation de l’entropie relative par méthode de monte-carlo. *C. R. Acad. Sci. Paris, Série I*, 332 :345–350, 2001.
- [LO97] R. Lagnado and S. Osher. A technique for calibrating derivative security pricing models : numerical solution of an inverse problem. *J. Comp. Fin.*, 1 :13–25, 1997.

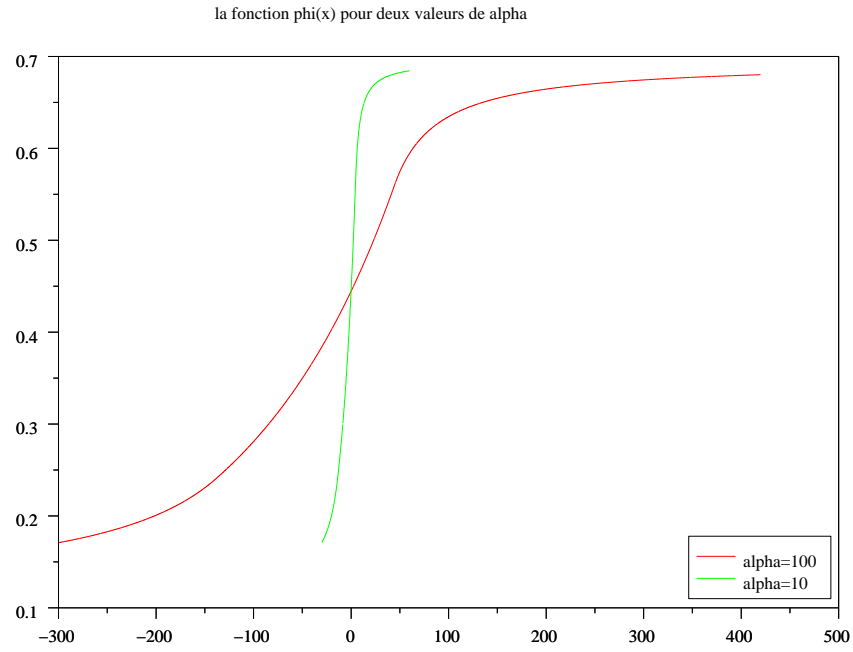


FIG. 1 –  $\Phi'(\Gamma)$  pour  $\alpha = 100$  et  $\alpha = 10$  .

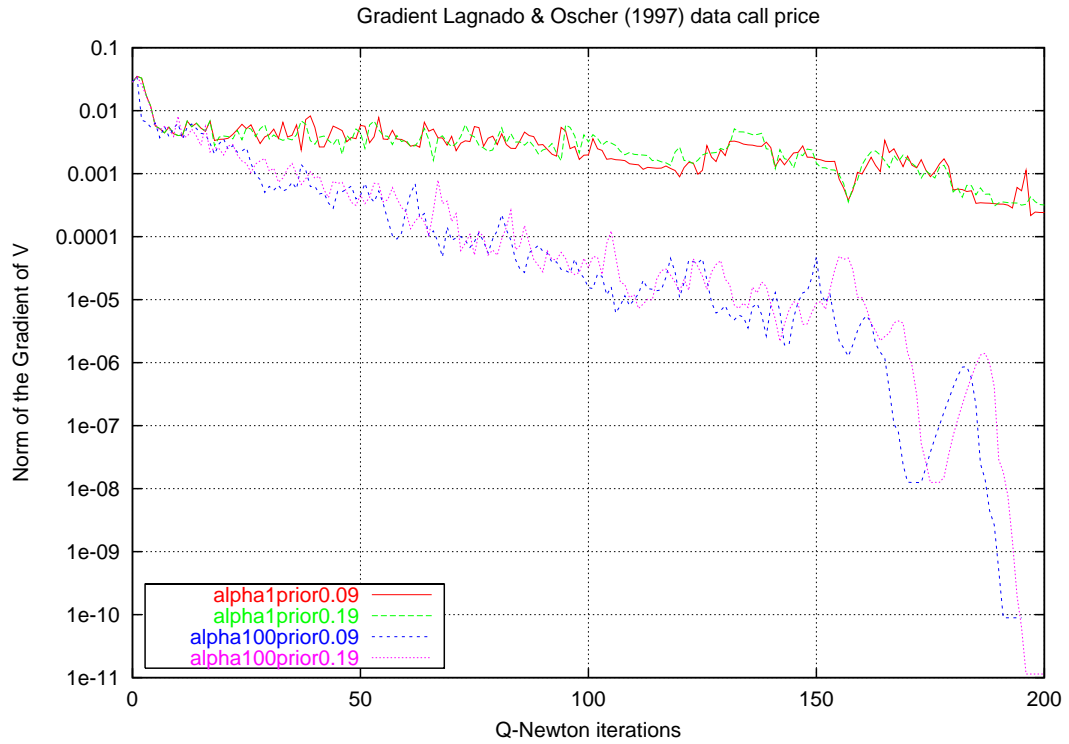


FIG. 2 – Gradient de  $V$  en fonction des itérations du Quasi-Newton pour différentes valeurs de  $\alpha$  et  $\sigma_0$ . Feuille de prix dans [LO97].

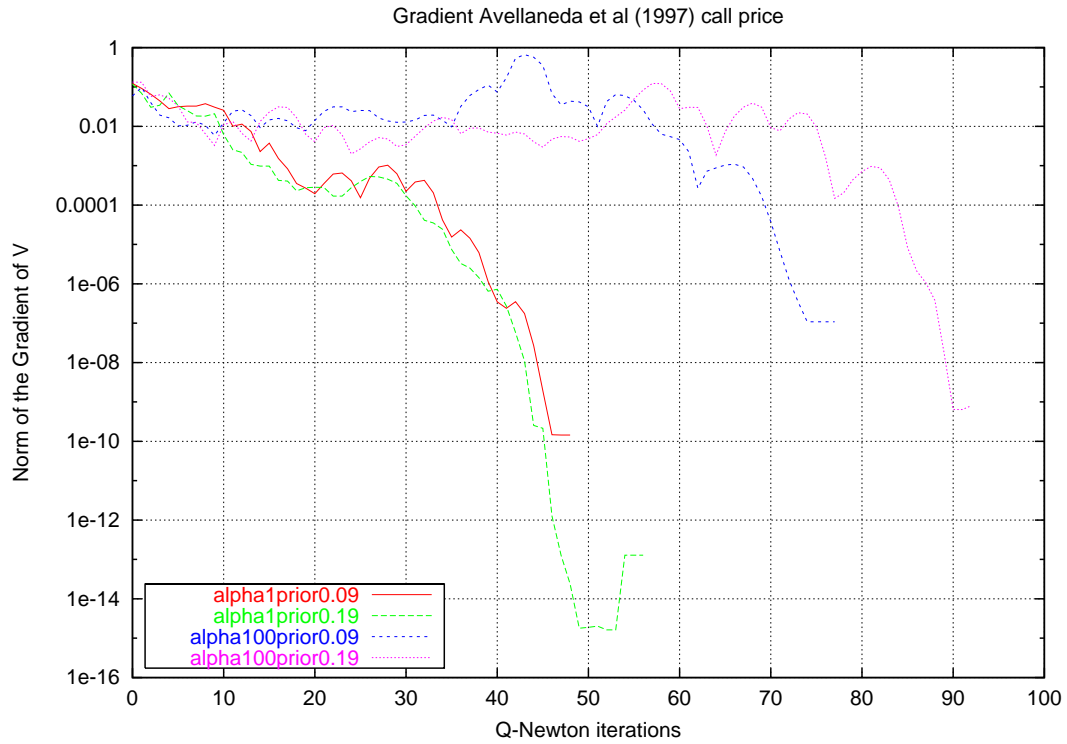


FIG. 3 – Gradient de  $V$  en fonction des itérations du Quasi-Newton pour différentes valeurs de  $\alpha$  et  $\sigma_0$ . Feuille de prix dans [AFHS97].

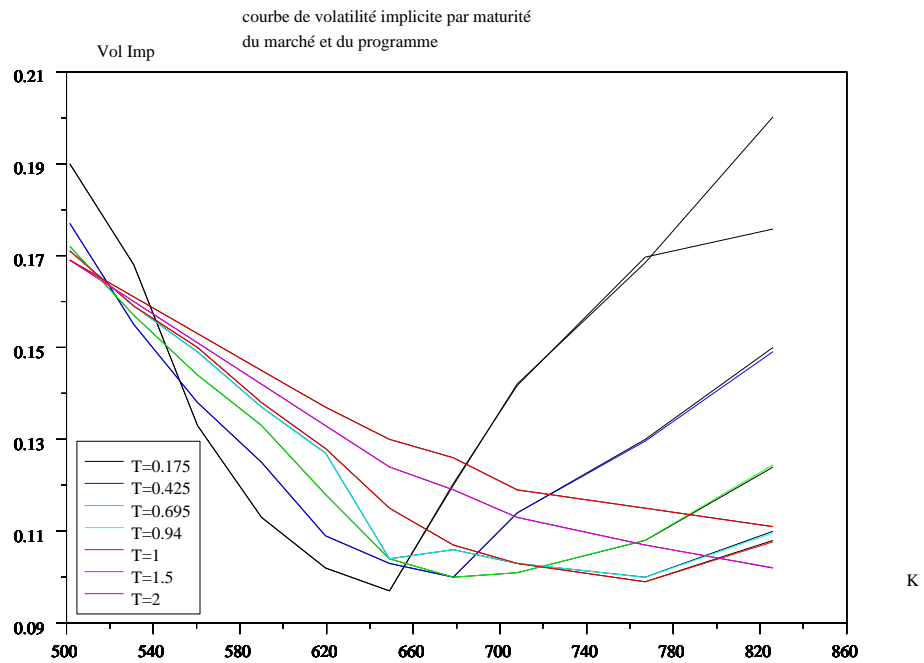


FIG. 4 – Volatilté implicite réelle et volatilté implicite de l'algorithme.

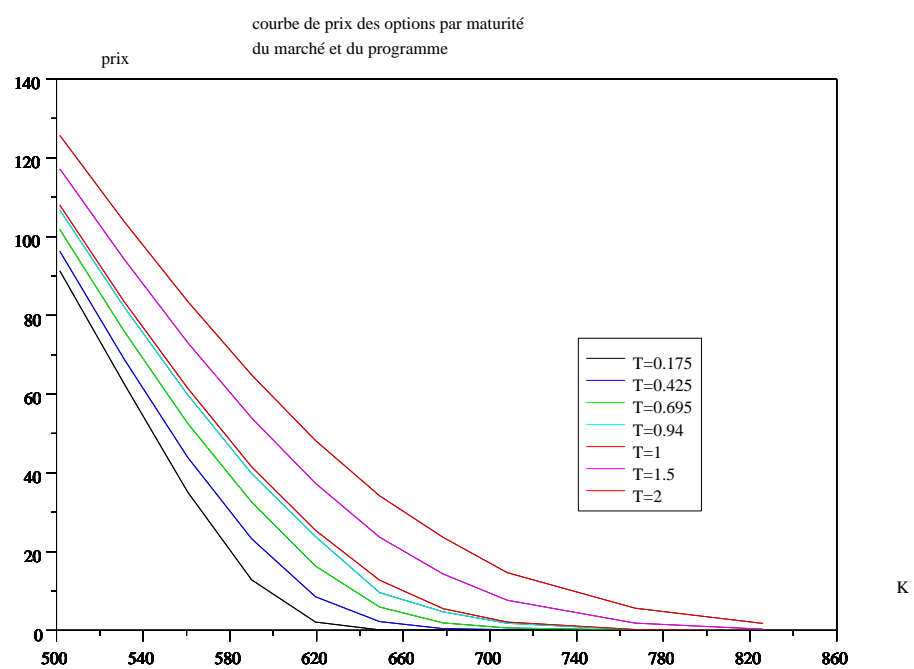


FIG. 5 – prix Réel et prix de l'algorithme par maturité.