UNIVERSITÉ PARIS - CERGY

CY Tech. Département Mathématiques

Option INGÉNIERIE FINANCIÈRE (MMF). Option ACTUARIAT.

Irina Kortchemski

MODEL CALIBRATION AND SIMULATION

TP4 Résolution numérique de l'équation de Dupire. Calibraion de volatilité locale dans le modèle de CEV.

Partie I L'équation de Dupire

On vous propose de résoudre numériquement l'équation de Dupire par la méthode implicite de Crank-Nicolson.

Prix d'une options Européene $V(T,K;S_0,t_0)$ porte sur un strike K et sur la maturité T. On fixe le prix initiale de l'actif S_0 et le temps t_0.

L'équation de Dupire en termes des variables K et T :

$$\frac{\partial V}{\partial T} + rK\frac{\partial V}{\partial K} - \frac{1}{2}\sigma^2_{locale}(K,T)K^2\frac{\partial^2 V}{\partial K^2} = 0$$

$\sigma_{locale}(K,T)$ est la volatilité locale. Prix de l'option à la maturité $T=0$

$$V(T=0,K;t_0,S_0) = max(S_0-K,0).$$

Maintenant on résout par les différences finies l'équation sur $[0,K_{max}]$ avec les conditions aux limites de Dirichlet :

$$\begin{cases} \frac{\partial V}{\partial T} + rK\frac{\partial V}{\partial K} - \frac{1}{2}\sigma^2_{locale}(K,T)K^2\frac{\partial^2 V}{\partial K^2} = 0 \\ V(T=0,K) = max(S_0-K,0) \\ V(T,K=0) = S_0 \\ V(T,K=K_{max}) = 0, \qquad K_{max}\to\infty \end{cases}$$

Théorie de discrétisation.

Discrétisons les variables : spatiale et temporelle :
$K : K_0 = K_{min} = 0, ...K_i, ...K_{N+1} = K_{max}$.
$T : T^0 = 0, ...T^n, ...T^{M+1} = T_{max}$.
Donc

$$\begin{cases} i = 0, 1, 2, ...N + 1 \\ n = 0, 1, 2, ...M + 1 \\ \Delta K = \frac{K_{max}}{N+1} \\ \Delta t = \frac{T_{max}}{M+1} \\ V(T^n, K_i) \equiv V_i^n \end{cases} \tag{1}$$

Conditions aux limites :

$$\begin{cases} V_0^n = S_0 \\ V_{N+1}^n = 0 \\ n = 0, 1, ...M \end{cases} \tag{2}$$

Conditions initiales :

$$\begin{cases} V_i^0 = max(S_0 - K_i, 0), i = 0, 1, 2..., N + 1 \end{cases} \tag{3}$$

Pour discrétiser l'équation (1) on utilise :
pour $\frac{\partial V}{\partial t}$ la dérivée avancée,
pour $\frac{\partial^2 V}{\partial K^2}$ la seconde dérivée, centrée,
pour $\frac{\partial V}{\partial K}$ la dérivée centrée.
On applique la méthode de Crank-Nicolson et on obtient l'équation de la forme :

$$B_i V_{i-1}^{n+1} + D_i V_i^{n+1} + A_i V_{i+1}^{n+1} = C_i^n,$$

$$A_i = \frac{\Delta t}{4}(\frac{K(i)}{\Delta K} r - \sigma_{loc}^2 (\frac{K(i)}{\Delta K})^2)$$

$$B_i = -\frac{\Delta t}{4}(\frac{K(i)}{\Delta K} r + \sigma_{loc}^2 (\frac{K(i)}{\Delta K})^2)$$

$$D_i = 1 + \frac{\Delta t}{2}\sigma_{loc}^2 (\frac{K(i)}{\Delta K})^2$$

$$C_i^n = -\frac{\Delta t}{4}(\frac{K(i)}{\Delta K} r - \sigma_{loc}^2 (\frac{K(i)}{\Delta K})^2) V_{i+1}^n + (1 - \frac{\Delta t}{2}\sigma_{loc}^2 (\frac{K(i)}{\Delta K})^2) V_i^n + \frac{\Delta t}{4}(\frac{K(i)}{\Delta K} r + \sigma_{loc}^2 (\frac{K(i)}{\Delta K})^2) V_{i-1}^n$$

On utilise les conditions aux limites

$$V_0^{n+1} = S_0, \quad V_{N+1}^{n+1} = 0$$

et on obtient la forme matricielle suivante :

$$\begin{pmatrix} D_1 & A_1 & 0 & 0 & 0 & & & \\ B_2 & D_2 & A_2 & 0 & 0 & & & \\ 0 & B_3 & D_3 & A_3 & 0 & & & \\ & & & & & B_{N-1} & D_{N-1} & A_{N-1} \\ & & & & & 0 & B_N & D_N \end{pmatrix} \cdot \begin{pmatrix} V_1^{n+1} \\ V_2^{n+1} \\ V_3^{n+1} \\ . \\ . \\ V_N^{n+1} \end{pmatrix} = \begin{pmatrix} C_1^n - B_1 S_0 \\ C_2^n \\ C_3^n \\ . \\ C_{N-1} \\ C_N^n \end{pmatrix}$$

Implementation en MATLAB

- Discretiser les valeurs du strike K :

$$K = linspace(0, K_{max}, N+2), \qquad \Delta K = K_{max}/(N+1)$$

Il y a $N+2$ composantes.

- Discrétiser les valeurs de la maturité T :

$$T = linspace(0, T_{max}, M+2), \qquad \Delta t = T_{max}/(M+1)$$

Il y a $M+2$ composantes.

- Dans l'algorithme de CN tout les indices **fixes** sont déjà deplacé de 1.

Algorithme de Crank-Nicolson

Pour $i = 1 : N+2$
$V(1,i) = max(S_0 - K_i, 0)$
Fin Pour

Pour $n = 2 : M+2$
$V(n,1) = S_0$
$V(n, N+2)) = 0$
Fin Pour

Pour $n = 1 : M+1$
Pour $i = 2 : N+1$

$$C_i^n = -\frac{\Delta t}{4}(\frac{K(i)}{\Delta K} r - \sigma_{loc}^2(\frac{K(i)}{\Delta K})^2)V_{i+1}^n + (1 - \frac{\Delta t}{2}\sigma_{loc}^2(\frac{K(i)}{\Delta K})^2)V_i^n + \frac{\Delta t}{4}(\frac{K(i)}{\Delta K} r + \sigma_{loc}^2(\frac{K(i)}{\Delta K})^2)V_{i-1}^n - \delta_2^i B_2 S_0$$

Fin Pour

$D_2^* = D_2,\ C_2^{*n} = C_2^n$

Pour $\quad i = 3 : N+1$
$D_i^* = D_i - \frac{B_i A_{i-1}}{D_{i-1}^*}, \quad C_i^{*n} = C_i^n - \frac{B_i C_{i-1}^{*n}}{D_{i-1}^*}$
Fin Pour

$V_{N+1}^{n+1} = \frac{C_{N+1}^{*n}}{D_{N+1}^*}$

Pour $\quad i = N : -1 : 2$
$V_i^{n+1} = \frac{C_i^{*n} - A_i V_{i+1}^{n+1}}{D_i^*}$
Fin Pour

Fin Pour

Utilisez les valeurs suivantes :

$$\begin{cases} K_{max} = 20 \\ S(0) = 10 \\ r = 0.1 \\ T_{max} = 0.5 \\ N = 199 \\ M = 49 \end{cases}$$

Travail à faire

1) Résolvez l'équation de Dupire et trouvez **la matrice** V_i^n.
Considerez deux cas pour la volatilité locale :

a) $\quad \sigma_l = 0.3$
b) $\quad \sigma_l = \sigma(T, K), \quad soit \quad \sigma(K_i) = \frac{\beta_1}{K_i^{\beta_2}}, \qquad \beta_1 = 1, \quad \beta_2 = 1$

Il est très utile de créer une fonction **Prix_Dupire**(h, β_1, β_2) de sorte :

function [V]=**Prix_Dupire**(h, β_1, β_2)
function$[\sigma]$ = **sigma_locale**$(\mathbf{h}, \mathbf{i})$
$\sigma(i) = \frac{\beta_1}{K(i)^{\beta_2}}$
end
Algorithme de Crank-Nicolson qui returne la matrice $V(n, i)$
end

2) Visualisez la fonction $V(T, K)$ pour les maturités T, $T/2$ et $T = 0$ en 2 dimensions et la surface $V(T, K)$ en 3 dimensions.

3) Important : Calculer la Vega de l'option :

$$\frac{\partial V(T, K, \sigma_l)}{\partial \sigma_l}$$

Pour cela calculer le prix $V(T,K,\sigma_l)$, puis le prix avec la volatilité légèrment dilatée : $V(T,K,\sigma_l+h)$, où $h=0.01$, puis Vega :

$$Vega(T,K,\sigma_l)=\frac{V(T,K,\sigma_l+h)-V(T,K,\sigma_l)}{h}, \qquad h=0.01$$

Il est très utile de créer une nouvelle fonction **Vega_Dupire** (h,β_1,β_2) de sorte :

function [vega]= **Vega_Dupire**(h,β_1,β_2)
vega=$\frac{\mathbf{Prix\_Dupire(h,\beta_1,\beta_2)-Prix\_Dupire(0,\beta_1,\beta_2)}}{h}$
end

La fonction **Prix_Dupire**(h,β_1,β_2) = calcule le prix de l'option pour la volatilité locale

$$\sigma_l+h=\frac{\beta_1}{K_i^{\beta_2}}+h$$

Dans l'espace discréte

$$Vega(n,i,\sigma_l)=\frac{V(n,i,\sigma_l+h)-V(n,i,\sigma_l)}{h}$$

5) Visualiser les fonctions de Vega pour $\sigma_l=0.3$ et $\sigma_l=\frac{1}{K}$ en 2 et 3 dimensions.

Partie II Calibration et la réconstruction de la volatilité locale.

Dans le cadre du modéle CEV (Constante Elasticity of Variance) on paramétrise la volatilité σ par des paramètres β_1 et β_2 :

$$\frac{dS_t}{S_t}=rdt+\sigma_{locale}(S,t)dW_t, \quad \sigma_{locale}(S,t)=\beta_1/S^{\beta_2}$$

L'estimation par l'algorithme de Levenberg-Marquardt porte sur les paramètres β_1 et β_2.
Nous allons chercher β_1 et β_2 pour minimiser la somme des carrés des résidus :

$$r_p=V(T_p,\mathcal{K}_p)^{marche}-V^{dupire}(T_p,\mathcal{K}_p,\sigma_{locale}(\beta_1,\beta_2)), \qquad \sigma_{locale}(\beta_1,\beta_2)=\beta_1/K^{\beta_2}.$$

Ici l'indice $p=1,2...,P$. P est la quantité des prix des options disponibles sur le marché.
Les valeurs du strike $\mathcal{K}_p$ sont les valeurs du tableau, les valeurs utiles.
La Jacobienne J_{pq} du vecteur des résidus par rapport aux paramètres inconnus β_q est une matrice $P\times 2$ dont la ligne p est

$$\frac{\partial r_p}{\partial\beta_1}=-\frac{\partial V_p^{dupire}}{\partial\sigma}\frac{\partial\sigma}{\partial\beta_1}=-\frac{\partial V_p^{dupire}(T_p,\mathcal{K}_p)}{\partial\sigma}\frac{1}{\mathcal{K}_p^{\beta_2}},$$

$$\frac{\partial r_p}{\partial\beta_2}=-\frac{\partial V_p^{dupire}}{\partial\sigma}\frac{\partial\sigma}{\partial\beta_2}=\frac{\partial V_p^{dupire}(T_p,\mathcal{K}_p)}{\partial\sigma}\frac{\beta_1\,ln(\mathcal{K}_p)}{\mathcal{K}_p^{\beta_2}}$$

$$J_{pq}(\beta^{(k)}) = \frac{\partial r_p}{\partial \beta_q^{(k)}},$$

où k est l'indice d'itération.

Le problème de calibration est un problème mal posé et il a besoin d'une régularisation. L'algorithme de Levenberg-Marquardt est une version régularisée de l'algorithme de Newton où la régularisation se fait par une matrice diagonale avec un coefficient λ : λD et assure l'inversement de la matrice $J^T J$.

On applique l'Algorithme de Levenberg-Marquardt :

$$(J^T J + \lambda D)\, d^{(k)} = -J^T r^{(k)} \quad \Rightarrow \quad d^{(k)} = -(J^T J + \lambda D)^{-1} J^T r^{(k)} \quad \Rightarrow \quad \beta^{(k+1)} = \beta^{(k)} + d^{(k)}$$

où D est une matrice diagonale positive.

Algorithme de calibration de Levenberg-Marquardt.

- Fonction à minimiser

$$I(\beta_1, \beta_2) = \sum_{p=1}^{6} \omega_p |\, V^{marche}(T_p, \mathcal{K}_p) - V^{dupire}(T_p, \mathcal{K}_p)(\sigma_{locale}(\beta_1, \beta_2))|^2$$

On choisit $\omega_p = 1$.

- Initialisation de l'algorithme : $T = 0.5 \quad \forall p$.

p	1	2	3	4	5	6	7	8
Strike $\mathcal{K}_p$	7	7.5	8	8.5	9	9.5	10	10.5
Prix d'option : $V_p^{marche}(\mathcal{K}_p)$	3.3634	2.9092	2.4703	2.0536	1.6666	1.3167	1.0100	0.7504

p	9	10	11	12	13	14	15
Strike $\mathcal{K}_p$	11	11.5	12	12.5	13	13.5	14
Prix d'option : $V_p^{marche}(\mathcal{K}_p)$	0.5389	0.3733	0.2491	0.1599	0.0986	0.0584	0.0332

Les valeurs du strike $\mathcal{K}_p$ sont les valeurs du tableau, les valeurs utiles.

Avant d'appliquer l'algorithme de Levenberg-Marquardt il faut extraire les prix utiles

$$V^{dupire}(T_p, \mathcal{K}_p, \sigma_{locale}(\beta_1, \beta_2))$$

de la matrice totale $V(n, i)$.

- Créer une fonction "Prix_Dupire_Utiles" qui choisit les prix utiles parmi tout les prix calculés et stockés dans la matrice V_i^n.

Pour cela on calcule les indices i_p qui correspond aux valeurs des strikes $\mathcal{K}_p$ du tableau, cotés sur le marché.

$$\mathcal{K}_p = \Delta K \cdot (i_p - 1) \Rightarrow i_p = \frac{\mathcal{K}_p}{\Delta K} + 1.$$

A la valeur $T = 0.5$ correspond l'indice $n = 51$.

- Calculer le Vega pour les valeurs des $\mathcal{K}_p$ utiles (Creer la fonction "Vega_Utiles") :

$$Vega(\mathcal{K}_p, T, \sigma^{(k)}) = \frac{\partial V}{\partial \sigma^{(k)}}(\mathcal{K}_p, T, \sigma^{(k)}), \quad p = 1, 2, ...15$$

- Choisir $\beta_1^{(0)} = 1, \quad \beta_2^{(0)} = 1$, la tolérence $\varepsilon = 0.00001, \ \lambda = 0.001, \quad k = 0.$

- Tant que le critère d'arret n'est pas satisfait $(\sqrt{(d_1^{(k)})^2 + (d_2^{(k)})^2} > \varepsilon)$ faire pour chaque iteration k :

- Calculer les résidues

$$r_p^{(k)} = V_p^{marche}(\mathcal{K}_p) - V^{dupire}(\mathcal{K}_p, T, \sigma(\beta_1^{(k)}, \beta_2^{(k)})), \qquad p = 1, 2, ...15$$

- Calculer le Jacobien :

$$\frac{\partial r_p^{(k)}}{\partial \beta_1^{(k)}} = -\frac{\partial V_p}{\partial \sigma} \frac{1}{\mathcal{K}_p^{\beta_2^{(k)}}} = Vega(\mathcal{K}_p, T, \beta_1^{(k)}, \beta_2^{(k)}) \frac{1}{\mathcal{K}_p^{\beta_2^{(k)}}},$$

$$\frac{\partial r_p^{(k)}}{\partial \beta_2^{(k)}} = \frac{\partial V_p}{\partial \sigma} \frac{ln(\mathcal{K}_p)\, \beta_1^{(k)}}{\mathcal{K}_p^{(\beta_2^{(k)})}} = -Vega(\mathcal{K}_p, T, \beta_1^{(k)}, \beta_2^{(k)}) \frac{ln(\mathcal{K}_p)\, \beta_1^{(k)}}{\mathcal{K}_p^{(\beta_2^{(k)})}}$$

- Calculer la matrice Hesienne

$$M = (J^T J + \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$$

- Calculer la direction de déscente

$$d = -M^{-1} J^T r$$

- Calculer

$$\beta_1^{(k+1)} = \beta_1^{(k)} + d_1^{(k)}, \quad \beta_2^{(k+1)} = \beta_2^{(k)} + d_2^{(k)}$$

- $k = k + 1$

Quels sont les paramètres β_1 et β_2 ?

Partie III. Modèle de Gatheral et la réconstruction de la volatilité locale.

Dans le cadre du modéle de Gatheral on paramétrise la volatilité σ_{locale} par des paramètres $a, \quad b, \quad \rho, \quad m.$

$$\frac{dS_t}{S_t} = rdt + \sigma_{locale}(S,t)dW_t, \quad \sigma_{locale}(S,t) = b(\rho(S-m) + \sqrt{(S-m)^2 + a^2})$$

On fixe $b = 0.05, \quad \rho = 0.1$. L'estimation par l'algorithme de Levenberg-Marquardt porte sur les paramètres $a = \beta_1$ et $m = \beta_2$.

Nous allons chercher β_1 et β_2 pour minimiser la somme des carrés des résidus :

$$r_p = V^{marche}(T_p, \mathcal{K}_p) - V^{dupire}(T_p, \mathcal{K}_p, \sigma_{locale}(\beta_1, \beta_2)), \qquad \sigma_{locale}(\beta_1, \beta_2) = b(\rho(\mathcal{K}_p - \beta_2) + \sqrt{(\mathcal{K}_p - \beta_2)^2 + \beta_1^2})$$

Ici l'indice $p = 1, 2..., P. \quad P = 14$ est la quantité des prix des options disponibles sur le marché.

$V^{marche}(T = 0.5, \mathcal{K}_p) = [5.2705 \quad 4.3783 \quad 3.5510 \quad 2.8138 \quad 2.1833$
$1.6651 \quad 1.2541 \quad 0.9374 \quad 0.6983 \quad 0.5195 \quad 0.3851 \quad 0.2817 \quad 0.1987 \quad 0.1277]$

$\mathcal{K}_p = [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]$

1) Tracer la surface de Dupire pour $a = 5, m = 5$.
2) Tracer la surface de Vega pour $a = 5, m = 5$.
3) Calibrer la volatilité locale.
4) Tracer le graphe de volatilité locale calibrée $K \to \sigma_{locale}(K, t)$
5) Tracer la surface de Dupire calibrée.
5) Tracer la surface de Vega calibrée.