

Le Mouvement Brownien Fractionnaire appliqué à la Log-Volatilité des actifs financiers (Rough Volatility)

RÉDA BENSAHLA - UNIVERSITÉ DE LILLE

September 23, 2021

Sommaire

1	Introduction	4
1.1	L'utilisation du mouvement brownien dans les modèles financiers modernes	4
1.2	Modélisation de la volatilité	4
1.3	Le mouvement brownien fractionnaire	5
1.3.1	Définition du mouvement brownien fractionnaire	5
1.3.2	Représentation de Mandelbrot-Van-Ness	5
1.3.3	Simulation du mouvement brownien fractionnaire	7
1.3.4	Quelques simulations du mouvement brownien fractionnaire	7
1.4	La volatilité rugueuse	8
1.4.1	Définition	8
1.4.2	Estimation de l'index de Hurst H	9
1.4.3	Application et estimation du paramètre de Hurst H sur d'autres indices	12
1.5	Équations Différentielles Stochastiques dirigées par un mBF	14
1.5.1	Présentation sous la forme différentielle et sous la forme intégrale	14
1.5.2	Schéma numérique de Milstein	14
1.6	Modèles et Processus construits sur le mouvement brownien fractionnaire	15
1.6.1	Linéarisation d'équations différentielles stochastiques unidimensionnelle	15
1.6.2	Présentation du processus d'Ornstein-Uhlenbeck fractionnaire	16
1.6.3	Solution théorique exacte	17
1.6.4	Modèle de Cox-Ingersoll-Ross fractionnaire	18
1.6.5	Le modèle de Black-Scholes fractionnaire	20
1.6.6	Modélisation d'un actif sous-jacent sous Black-Scholes fractionnaire	20
1.7	Modèle Rough Heston	21
1.7.1	Modèle de Heston classique	21
1.7.2	Modèle de Heston dirigé par un mouvement brownien fractionnaire	21
1.7.3	Construction du modèle Rough Heston	22
1.7.4	Démonstration de la loi du couple $(S_t^{t_0}, V_t^{t_0})$ conditionnellement à F_t	24
1.7.5	Schéma numérique de calcul de la fonction caractéristique du log-prix dans le modèle Rough Heston	26
1.8	Le modèle rough SABR	28
1.8.1	Le modèle SABR classique	28
1.8.2	Le modèle SABR dirigé par un mouvement brownien fractionnaire	29

1.9 Le modèle RFSV	29
Conclusion	30
A L'approche de Carr et Madan	30
B Code Python pour simulations et estimations	31
B.1 Code Python pour l'estimation du paramètre de Hurst H	32
B.2 Code Python pour les simulations du mouvement brownien fractionnaire et du processus d'Ornstein-Uhlenbeck	38
B.3 Code Python pour les simulations du Modèle de Black-Scholes fractionnaire et du modèle Cox-Ingersoll-Ross	43
C Bibliographie	48

1 Introduction

1.1 L'utilisation du mouvement brownien dans les modèles financiers modernes

Dans la modélisation financière moderne, les prix ont toujours été modélisés par des martingales semi-continues, de plus, il a toujours été nécessaire de présenter certaines hypothèses, qui aujourd'hui peuvent être remises en cause, notamment :

- L'hypothèse de normalité des rendements, ceux-ci présentent en général des asymétries et des queues épaisses.
- La continuité des cours, pour cela il suffit d'observer les sauts, plus communément appelés "gaps" entre deux instants de quotations de cours.
- Mais surtout, et le plus important, l'hypothèse d'indépendance des accroissements, en effet, il s'agit ici d'ignorer l'impact des événements réalisés dans le passé, pour cela, Mandelbrot préconise d'utiliser le mouvement Brownien Fractionnaire à la place du mouvement Brownien.

C'est donc ce point qui nous a amené à faire cette étude sur le mouvement Brownien Fractionnaire, de plus, la Log-Volatilité étant fractionnaire l'utilisation de ce mouvement Brownien nous donne donc des modèles assez consistant, cependant, comme nous le verrons par la suite, il est assez difficile à mettre en place.

1.2 Modélisation de la volatilité

Les Logs-prix sont souvent modélisés comme des semi-martingales continues. Pour un actif donné avec un Log-Prix Y_t , ce dernier peut être modélisé comme suit :

$$dY_t = \mu_t dt + \sigma_t dW_t$$

où μ_t est le terme de drift et W_t est un mouvement brownien uni-dimensionnel. Le terme σ_t représente quant à lui le processus de volatilité du modèle. D'un côté nous avons des modèles comme le modèle de Black-Scholes où la volatilité est souvent constante ou est une fonction déterministe du temps, puis d'un autre côté, nous avons des modèles à volatilité stochastiques, la volatilité σ_t est modélisée par une semi-martingale brownienne, parmi ces modèles nous pouvons citer le modèle de Heston dont voici la dynamique de la volatilité :

$$d\nu_t = \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t}dB_t$$

Ou encore le modèle CEV ("Constant Elasticity of Variance Model") dont la volatilité est stochastique mais déterministe, sa dynamique s'écrit comme suit :

$$dS_t = \mu S_t dt + \sigma S_t^\gamma dB_t$$

1.3 Le mouvement brownien fractionnaire

1.3.1 Définition du mouvement brownien fractionnaire

Dans cette section, nous allons définir le mouvement brownien fractionnaire, ses propriétés, comment le simuler et enfin comment estimer l'index de Hurst (H).

Le mouvement brownien fractionnaire noté $\{B^H(t)\}_{t \in \mathbb{R}}$ d'exposant de Hurst $H \in]0; 1]$, est l'unique processus gaussien centré, nul en zéro, dont les accroissements sont stationnaires et auto-similaires, il est défini par sa fonction de covariance $\Gamma(t, s)$:

$$\Gamma(t, s) = \mathbb{E}[B^H(s)B^H(t)] = \frac{C(H)}{2}(|t|^{2H} + |s|^{2H} + |t-s|^{2H})$$

Sa fonction d'auto-covariance $\gamma(t, s)$ est donnée par :

$$\gamma(t, s) = \frac{C(H)}{2}(|t-s-1|^{2H} - 2|t-s|^{2H} + |t-s+1|^{2H})$$

où

$$C(H) = \text{Var}[B^H(1)]$$

Où : $B^H(0) = 0$ et $B(s)$ représente le mouvement brownien standard
Lorsque $C(H) = 1$ ce processus est appelé Mouvement Brownien Fractionnaire standard, de plus lorsque $H = \frac{1}{2}$, le mouvement brownien fractionnaire correspond à un mouvement brownien standard

1.3.2 Représentation de Mandelbrot-Van-Ness

Une autre forme de représentation du mouvement brownien fractionnaire est celle de Mandelbrot-Van-Ness, aussi appelé moving-average representation, celle-ci est donnée par :

$$B^H(t) := \frac{1}{C(H)} \int_{\mathbb{R}} [((t-s)^+)^{H-\frac{1}{2}} - ((-s)^+)^{H-\frac{1}{2}}] dB(s),$$

$$\text{Avec : } C(H) = \left(\int_0^\infty \left((1+s)^{H-\frac{1}{2}} - s^{H-\frac{1}{2}} \right)^2 ds + \frac{1}{2H} \right)^{\frac{1}{2}}$$

Nous allons démontrer ce théorème :

On définit $f_t(s) = ((t-s)^+)^{H-\frac{1}{2}} - ((-s)^+)^{H-\frac{1}{2}}$, $s \in \mathbb{R}, t \geq 0$

Il est nécessaire de souligner que dans la définition de l'intégrale de Wiener l'opérateur intégrant doit être dans L^2 , autrement dit : $\int_{\mathbb{R}} f_t(s)^2 ds < \infty$,

Lorsque $H \neq \frac{1}{2}$, $s \rightarrow -\infty$, $f_t(s)$ tends vers $(-s)^{H-\frac{3}{2}}$, de carré intégrable, et pour $t \geq 0$ posons :

$$X_t = \int_{\mathbb{R}} \left[((t-s)^+)^{H-\frac{1}{2}} - ((-s)^+)^{H-\frac{1}{2}} \right] dB_s$$

On a par l'isométrie d'Itô :

$$\mathbb{E}[X_t^2] = \int_{\mathbb{R}} \left[((t-s)^+)^{H-\frac{1}{2}} - ((-s)^+)^{H-\frac{1}{2}} \right]^2 ds$$

Pour $t > 0$, le résultat étant trivial pour $t = 0$, en factorisant par t , on a :

$$= t^{2H} \int_{\mathbb{R}} \left[((1-u)^+)^{H-\frac{1}{2}} - ((-u)^+)^{H-\frac{1}{2}} \right]^2 du$$

On décompose l'intégrale sur \mathbb{R}^- et sur \mathbb{R}^+ , on a :

$$\begin{aligned} &= t^{2H} \left(\int_{-\infty}^0 [(1-u)^{H-\frac{1}{2}} - (-u)^{H-\frac{1}{2}}]^2 du + \int_0^1 (1-u)^{2H-1} du \right) \\ &= C(H)^2 t^{2H} \end{aligned}$$

On a aussi, pour $s < t$:

$$\begin{aligned} \mathbb{E}[|X_t - X_s|^2] &= \int_{\mathbb{R}} \left[((t-u)^+)^{H-\frac{1}{2}} - ((s-u)^+)^{H-\frac{1}{2}} \right]^2 du \\ \mathbb{E}[|X_t - X_s|^2] &= \int_{\mathbb{R}} \left[((t-s-u)^+)^{H-\frac{1}{2}} - ((-u)^+)^{H-\frac{1}{2}} \right]^2 du \\ \mathbb{E}[|X_t - X_s|^2] &= C(H)^2 |t-s|^{2H} \end{aligned}$$

Malheureusement, comme nous allons le montrer par la suite, le mouvement brownien fractionnaire n'est pas une semi-martingale pour $H \neq \frac{1}{2}$, on ne peut donc pas lui appliquer toute la théorie d'Itô sur l'intégration stochastique des semi-martingales

Démontrons cette précédente affirmation, l'idée étant de montrer que si le mouvement brownien fractionnaire est une semi-martingale alors la variation quadratique doit être finie. : Posons $p > 0$:

$$Y_{n,p} = n^{pH-1} \sum_{j=1}^n |B_{j/n} - B_{(j-1)/n}|^p$$

Par la propriété d'auto-similarité du mouvement brownien fractionnaire, la suite $(Y_{n,p}, n \geq 1)$ a la même distribution que $(\tilde{Y}_{n,p}, n \geq 1)$, avec :

$$\tilde{Y}_{n,p} = n^{-1} \sum_{j=1}^n |B_j - B_{j-1}|^p$$

Par le théorème ergodique on sait que $\tilde{Y}_{n,p}$ converge presque sûrement vers $\mathbb{E}[|B_1|^p]$ quand n tend vers l'infini, donc $Y_{n,p}$ converge en probabilité quand n tend vers l'infini vers $\mathbb{E}[|B_1|^p]$, et on a donc :

$$V_{n,p} = \sum_{j=1}^n |B_{j/n} - B_{(j-1)/n}|^p$$

Converge en probabilité vers 0 quand n tend vers l'infini si $pH < 1$, il est nécessaire de distinguer deux cas :

- Si $H < \frac{1}{2}$, prenons $p > 2$ tel que $pH < 1$ ce qui a pour conséquence que la p -variation du mouvement brownien fractionnaire est infini, pour $p = 2$ la variation quadratique est aussi finie.
- Si $H > \frac{1}{2}$, on peut choisir p tel que $\frac{1}{H} < p < 2$, alors la p -variation vaut 0, et, en conséquence, la variation quadratique est aussi 0.
De plus, si on choisit p tel que $1 < p < \frac{1}{H}$, on déduit que la variation quadratique totale est infinie.

Ainsi, nous avons démontré que pour $H \neq \frac{1}{2}$ le mouvement brownien fractionnaire ne peut donc pas être une semi-martingale.

Le comportement du mouvement brownien fractionnaire dépend donc de ce paramètre H :

- Lorsque $H < \frac{1}{2}$ les trajectoires sont assez irrégulières (comparativement au mouvement brownien standard).
- Lorsque $H > \frac{1}{2}$ les trajectoires sont plus régulières, on constate l'effet de longue mémoire.

1.3.3 Simulation du mouvement brownien fractionnaire

Nous allons donc présenter ici une méthode de simulation du mouvement brownien fractionnaire : La méthode de Cholesky

Méthode de Cholesky Il s'agit ici d'une méthode qui est exacte en théorie, mais qui n'est malheureusement pas très utile en pratique car son exécution nécessite beaucoup de temps.

Posons Γ la matrice de covariance du mouvement brownien discrétisé aux instants $\frac{i}{n}$, pour $i = 0, \dots, n-1$.

La méthode de Cholesky consiste à déduire Γ' de Γ en supprimant la première ligne et la première colonne, la matrice Γ' est symétrique définie positive et admet donc une décomposition de Cholesky $\Gamma' = LL^t$ où L est une matrice triangulaire inférieure.

Par la suite, on effectue le produit matriciel LZ où Z est un vecteur de $n-1$ variables aléatoires indépendantes gaussiennes centrées et réduite, le vecteur LZ est un vecteur gaussien centré et on a : $\mathbb{E}[(LZ)(LZ)^t] = \Gamma'$

Ainsi, le vecteur $B_H = (0, (LZ)^t)^t$ représente une trajectoire du mouvement brownien discrétisé.

1.3.4 Quelques simulations du mouvement brownien fractionnaire

...

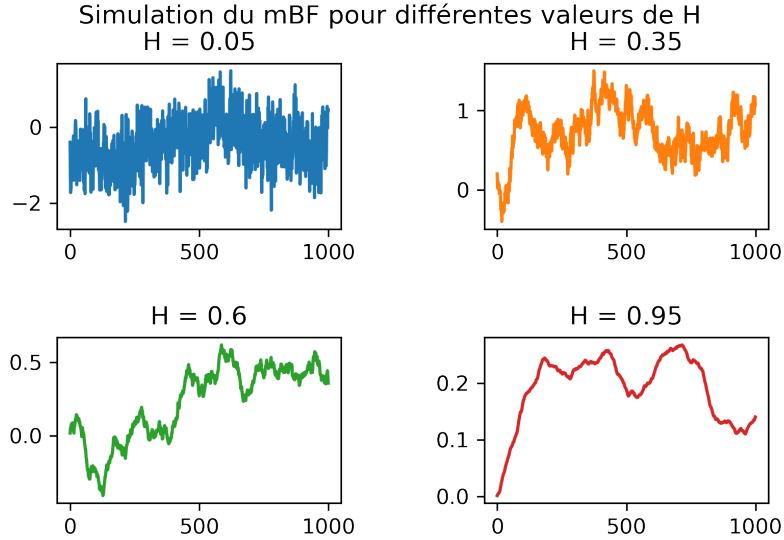


FIGURE 1 – Simulation du mouvement brownien fractionnaire pour différentes valeurs de H

1.4 La volatilité rugueuse

1.4.1 Définition

Les différents modèles historiques dans la modélisation financière moderne proposent deux approches concernant le processus de volatilité σ_t :

- Premièrement, des trajectoires assez régulières (constantes ou déterministes), dans le cas du modèle de Black-Scholes standard.
- Deuxièmement, dans le cas des modèles à volatilité locales ou stochastiques, comme le modèle de Heston, les trajectoires de volatilité modélisées correspondaient pratiquement à celles du mouvement brownien standard.

Dans l'article de Comte et Renault, les auteurs proposent, en partant du fait que la volatilité est un processus à mémoire-longue, de modéliser la log-volatilité en utilisant un mouvement brownien fractionnaire avec pour paramètre $H \in]\frac{1}{2}, 1[$, ce modèle est appelé Fractional Stochastic Volatility (FSV)

Dans l'article de Gatheral, Jaisson et Rosenbaum, un modèle plus complexe est proposé, le Rough Fractional Stochastic Volatility (RFSV), celui-ci est le même que le modèle FSV mais en prenant $H \in]0; \frac{1}{2}[$, nous détaillerons ces modèles dans la section ().

Nous utiliserons par la suite le mouvement brownien fractionnaire afin de modéliser les incréments de la log-volatilité, en partant du principe que les incréments stationnaires du mouvement brownien fractionnaire satisfont, pour tout $t \in \mathbb{R}, \Delta \geq 0, q > 0$:

$$\mathbb{E}[|B_{t+\Delta}^H - B_t^H|^q] = K_q \Delta^{qH}$$

Lorsque $H > \frac{1}{2}$, les incréments du mouvement brownien fractionnaire sont positivement corrélés et montre un effet de mémoire longue, en effet, la trajectoire de B_{t+1}^H dépend de B_t^H , et on a donc :

$$\sum_{k=0}^{+\infty} Cov[W_1^H, W_k^H - W_{k-1}^H] = +\infty$$

En effet, $Cov[W_1^H, W_k^H - W_{k-1}^H]$ est de l'ordre k^{2H-2} avec $k \rightarrow \infty$

Où K_q représente le moment d'ordre q de la valeur absolue d'une variable gaussienne centrée et réduite ; Δ représente la variation temporelle.

L'approche de Gatheral et Rosenbaum est de partir du principe que l'on a accès à différentes observations du processus de volatilité σ_t et de découper les instants de volatilités sur $[0; T]$ comme suit : $\sigma_0, \dots, \sigma_\Delta, \dots, \sigma_{k\Delta}, \dots : k \in [0, N]$, où $N = \lfloor T/\Delta \rfloor$.

On définit donc, pour $q \geq 0$:

$$m(q, \Delta) = \frac{1}{N} \sum_{k=1}^N |\log(\sigma_{k\Delta}) - \log(\sigma_{(k-1)\Delta})|^q$$

1.4.2 Estimation de l'index de Hurst H

La méthode d'estimation du paramètre H proposée par Gatheral, Jaisson et Rosenbaum se construit comme suit :

Il faut comprendre le comportement de la quantité $\frac{\log m(q, \Delta)}{\log(\Delta)}$, on s'aperçoit que cette relation est linéaire, à un coefficient ζ_q près, on s'aperçoit (cf Figure 2) que pour tout q : $m(q, \Delta) \propto \Delta^{\zeta_q}$

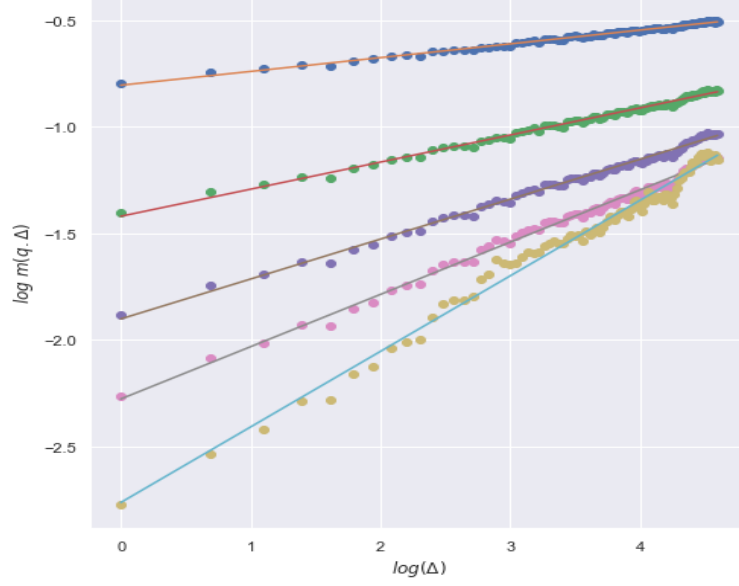


FIGURE 2 – Relation entre $\log(m(q, \Delta))$ et $\log(\Delta)$; Calculé sur l'indice CAC 40

Enfin il s'agit d'étudier le comportement de ζ_q en fonction de q , nos résultats nous montrent qu'il existe donc une relation linéaire entre ζ_q et q , de ce résultat est déduit la pente la droite qui correspond donc au paramètre H (cf Figure 3), et on déduit donc la relation d'échelle monofractal suivante :

$$\zeta_q = qH$$

Il convient de remarquer que H varie très peu en fonction du temps.

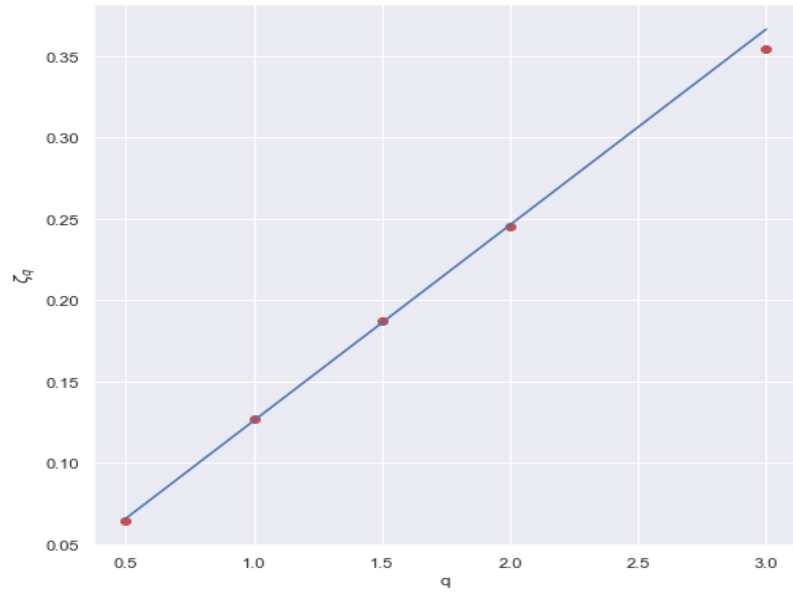


FIGURE 3 – Relation entre ζ_q et q

1.4.3 Application et estimation du paramètre de Hurst H sur d'autres indices

On applique la même méthode sur différentes volatilités de différents indices, les résultats sont les suivants :

	Indices	Estimation du paramètre H	Estimation du paramètre ν
0	.FCHI	0.122568	0.320464
1	.AEX	0.133145	0.320414
2	.STOXX50E	0.100548	0.397587
3	.BFX	0.125937	0.306290
4	.IBEX	0.112744	0.314685
5	.GDAXI	0.122359	0.331459
6	.AORD	0.071875	0.459212
7	.FTSE	0.109180	0.365529
8	.MXX	0.079091	0.414202
9	.IXIC	0.123072	0.350766
10	.SPX	0.131715	0.381304
11	.RUT	0.110748	0.387228
12	.SSMI	0.149400	0.278083
13	.DJI	0.123347	0.390174
16	.KS11	0.102043	0.319863
17	.BVSP	0.127826	0.333823
18	.HSI	0.084487	0.327221
19	.KSE	0.099285	0.444501
20	.N225	0.105592	0.372272
21	.SSEC	0.105825	0.360319
22	.OSEAX	0.107394	0.369890
23	.GSPTSE	0.110107	0.391804
24	.MSI	0.109951	0.335748
25	.OMXHPI	0.107753	0.356045
26	.OMXSPI	0.116433	0.354460
27	.OMXC20	0.098222	0.364126
29	.FTMIB	0.114010	0.345525

FIGURE 4 – Estimation du paramètre H de différents indices

On remarque que le paramètre H varie entre 0.071 et 0.149, de plus on remarque que la variation de l'estimation du paramètre H depuis la simulation

effectuée par Rosenbaum et Gatheral (En Mai 2016, cf Figure 5) n'a que très peu variée, ce qui confirme l'affirmation de la constance du paramètre H dans le temps. Enfin, cela permet aux praticiens d'éviter à estimer H souvent et contrairement à d'autres problématiques qui nécessitent de réajuster le calibrage tous les jours (Par exemple la calibrage de la volatility smile).

	h_est	names	nu_est
0	0.133954	SPX2.rk	0.321337
1	0.142315	FTSE2.rk	0.270677
2	0.113366	N2252.rk	0.320396
3	0.150251	GDAXI2.rk	0.274873
4	NaN	RUT2.rk	NaN
5	0.083370	AORD2.rk	0.359025
6	0.131013	DJI2.rk	0.317327
7	NaN	IXIC2.rk	NaN
8	0.130485	FCHI2.rk	0.291967
9	0.103370	HSI2.rk	0.281453
10	0.125847	KS11.rk	0.274713
11	0.145671	AEX.rk	0.290187
12	0.178427	SSMI.rk	0.223249
13	0.128117	IBEX2.rk	0.281662
14	0.110092	NSEI.rk	0.324883
15	0.092252	MXX.rk	0.324180
16	0.106595	BVSP.rk	0.312907
17	NaN	GSPTSE.rk	NaN
18	0.119857	STOXX50E.rk	0.337045
19	0.127094	FTSTI.rk	0.228910
20	0.133361	FTSEMIB.rk	0.298712

FIGURE 5 – Estimation du paramètre H de différents indices, en Mai 2016, par Gatheral et Rosenbaum

1.5 Équations Différentielles Stochastiques dirigées par un mBF

Dans cette section nous allons nous intéresser à la résolution théorique de certaines EDS ou lorsque la solution théorique n'est pas atteignable, nous présenterons un schéma de résolution numérique (Schéma de Milstein). Enfin, nous parlerons des équations dirigées par le mouvement brownien fractionnaire tels que le processus d'Ornstein-Uhlenbeck, le modèle CIR ou encore le modèle de Black Scholes fractionnaire.

1.5.1 Présentation sous la forme différentielle et sous la forme intégrale

L'équation différentielle stochastique dirigée par un mouvement brownien fractionnaire est donnée par :

$$dY_t = f(Y_t, t)dt + g(Y_t, t)dB_t^H$$

Où B_t^H est un mouvement brownien fractionnaire standard.

Une présentation alternative, sous forme d'intégrale :

$$Y_t = Y_s + \int_s^t f(Y_u, u)du + \int_s^t g(Y_u, u)dB_u^H \quad (1)$$

Pour le cas où $H = \frac{1}{2}$ l'intégrale stochastique de l'équation (1) est une martingale locale, ainsi le processus Y_t est, au minimum, une semi-martingale, ainsi la formule d'Itô est applicable.

Cependant, dans le cas où $H \neq \frac{1}{2}$ il est nécessaire de discuter les cas en fonction de H :

- Dans le cas où $H > \frac{1}{2}$ sous certaines conditions de régularité, on peut définir l'intégrale de Stieltjes.
- Dans le cas où $H \in]\frac{1}{4}; \frac{1}{2}[$ on pourrait définir une notion de chemins rugueux pour définir des intégrales stochastiques fractionnaires.

1.5.2 Schéma numérique de Milstein

Une EDS ne possède pas souvent une solution théorique explicite (formule fermée), et il est donc parfois nécessaire d'avoir recours à des Schémas numérique de calcul, dans le cas où nous avons une EDS dirigée par un mouvement brownien standard, il peut être judicieux d'utiliser un Schéma numérique d'Euler.

Le Schéma de Milstein est valable pour tout $H \in]0; 1[$, sa formule est donnée par :

$$\begin{aligned} \chi(t_{i+1}) = & \chi(t_i) + f(\chi(t_i), t_i)\Delta t + g(\chi(t_i), t_i)\Delta B_i^H \\ & + \frac{1}{2}g(\chi(t_i), t_i)g'(\chi(t_i), t_i)[(\Delta B_i^H)^2 - (t_{i+1}^{2H} - t_i^{2H})] + \epsilon_i \end{aligned} \quad (1)$$

ϵ_i représente un résidu (différence entre la solution théorique et la solution numérique donnée par la solution numérique de Milstein)

On peut donner une version plus courte et plus pratique de la formule, en partant du principe que : $t_{i+1}^{2H} = (t_i + \Delta t) \approx t_i^{2H} + 2Ht_i^{2H-1}\Delta t$

Ce qui nous donne donc :

$$\begin{aligned} \chi(t_{i+1}) = & \chi(t_i) + f(\chi(t_i), t_i) - g(\chi(t_i), t_i)g'(\chi(t_i), t_i)Ht_i^{2H-1}\Delta t \\ & + g(\chi(t_i), t_i)\Delta B_i^H + \frac{1}{2}g(\chi(t_i), t_i)g'(\chi(t_i), t_i)(\Delta B_i^H)^2 + \epsilon_i' \end{aligned} \quad (2)$$

ϵ_i' représente le résidu, et est supérieur à ϵ_i car l'approximation de t_{i+1}^{2H} accroît cette marge d'approximation.

1.6 Modèles et Processus construits sur le mouvement brownien fractionnaire

1.6.1 Linéarisation d'équations différentielles stochastiques unidimensionnelle

Pour pouvoir avancer dans la suite de notre étude (pour pouvoir donner une solution exacte du Processus d'Ornstein-Uhlenbeck fractionnaire) nous avons besoin d'introduire, suivant la présentation de [UD07], une méthode de construction de solutions théoriques.

Théorème 1 Posons :

$$dX_t = f(X_t, t)dt + g(X_t, t)dB_t^H \quad (1)$$

Où $X_0 = x_0, t \geq 0, H \in (0, 1)$ et dB_t^H l'incrément du mouvement brownien fractionnaire B_t^H , avec f et g , deux fonctions supposées assez régulières, l'équation (1) peut s'écrire sous la forme linéaire suivante :

$$dY_t = (a(t)Y_t + b(t))dt + (c(t)Y_t + e(t))dB_t^H \quad (2)$$

Via la transformation inverse $y = h(x, t)$ avec sa dérivée partielle par rapport à x non-nulle si et seulement si on a :

$$\frac{\partial}{\partial x} \left(\frac{\frac{\partial}{\partial x}(g(x, t)) \frac{\partial}{\partial x}(g(x, t)L)}{\frac{\partial}{\partial x}(g(x, t)L)} \right) = 0$$

Avec $L = \frac{\partial}{\partial x} \left(\frac{1}{g} \right) + \frac{\partial}{\partial x} \left(\frac{f}{g} - Ht^{2H-1} \frac{\partial g}{\partial x} \right)$

Les étapes de la linéarisation sont les suivantes, et dépendent de $c(t)$ de l'équation (2) :

— Si $c(t) = 0$ alors on a :

$$h(x, t) = \left(\int \alpha(t) dt \right)^{-1} \left(\int \frac{1}{g(x, t)} dx \right)$$

La condition $\frac{\partial}{\partial x} \left(\frac{\frac{\partial}{\partial x}(g(x, t)) \frac{\partial}{\partial x}(g(x, t)L)}{\frac{\partial}{\partial x}(g(x, t)L)} \right) = 0$ est bien respectée et l'équation différentielle stochastique linéaire s'écrit donc :

$$dY_t = \beta(t)e(t)dt + e(t)dB_t^H$$

avec $e(t) = \left(\int \alpha(t) dt \right)^{-1}$, α et β sont définies par identification dans l'égalité :

$$\alpha(t)y + \beta(t) = \int \frac{\partial}{\partial t} \left(\frac{1}{g} \right) + \frac{f}{g} - Ht^{2H-1} \frac{\partial g}{\partial x}$$

— Si $c(t) \neq 0$ alors on a :

$$h(x, t) = e^{-M(t) \int \frac{1}{g(x, t)} dx}$$

$$\text{Avec } M(t) = \frac{\frac{\partial}{\partial x}(g(x, t)) \frac{\partial}{\partial x}(g(x, t)L)}{\frac{\partial}{\partial x}(g(x, t)L)}$$

Nous avons maintenant une forme linéaire de notre équation, il faut maintenant la résoudre en utilisant la méthode du facteur intégrant, définit comme suit :

Définition 1 La fonction $F = F(t, B_t^H)$ vérifiant $d(Fy) = D_1(t, B_t^H)dt + D_2(t, B_t^H)dB_t^H$ est un facteur intégrant pour des équations différentielles stochastiques linéaires unidimensionnelles, on peut donc écrire :

$$F(t, B_t^H) = \exp \left(- \int^t c(s)dB_s^H + H \int^t s^{2H-1}c^2(s)ds - \int^t a(s)ds \right)$$

Par suite, on a donc la solution de l'équation différentielle stochastique linéaire suivante :

$$Y_t = \frac{1}{F} \left(\int^t F(s)b(s)ds - 2H \int^t s^{2H-1}F(s)c(s)e(s)ds + \int^t F(s)e(s)dB_s^H + y_0 \right)$$

1.6.2 Présentation du processus d'Ornstein-Uhlenbeck fractionnaire

Le processus de Ornstein-Uhlenbeck associé au mouvement brownien fractionnaire associé est donné par l'équation suivante :

$$dX_t = \theta(t)(\mu(t) - X_t)dt + \sigma(t)dB_t^H$$

Ce processus est souvent utilisé pour modéliser des taux en finance.

1.6.3 Solution théorique exacte

Pour donner une solution théorique exacte on utilise la méthode du facteur intégrant précédemment introduit (cf section 1.6.1) dont l'expression est la suivante : $F(t) = e^{\int^t \theta(s) ds}$

On a donc la solution suivante :

$$Y_t = e^{\int^t \theta(s) ds} \left(\int^t e^{\int^s \theta(s) ds} \theta(s) \mu(s) ds + \int^t e^{\int^s \theta(s) ds} \sigma(s) dB_s^H + y_0 \right) \quad (3)$$

Il s'agit maintenant de donner une solution numérique à (3). Pour cela, on utilise le schéma de Milstein et on a donc, pour le paramètre de Hurst H quelconque :

$$X_{j+1} = X_j + \theta(\mu - X_j)\Delta t + \sigma \Delta B_j^H$$

Quelques simulations du processus d'Ornstein-Uhlenbeck fractionnaire ...

Simulation du processus d'Ornstein-Uhlenbeck fractionnaire pour différentes valeurs de H

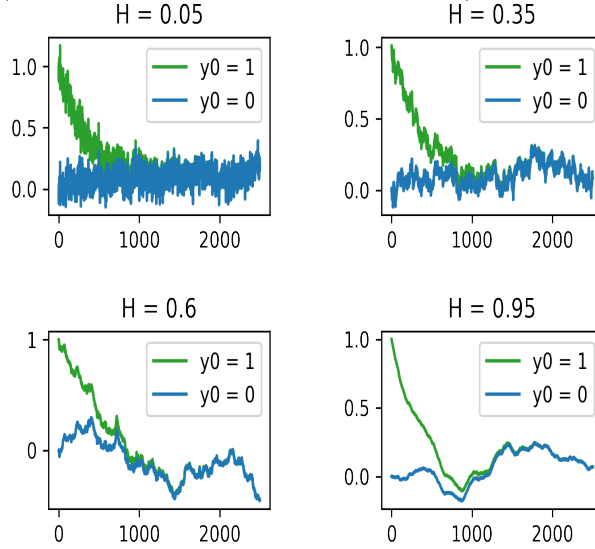


FIGURE 6 – Simulation du processus d'Ornstein-Uhlenbeck pour différentes valeurs de H

1.6.4 Modèle de Cox-Ingersoll-Ross fractionnaire

Le modèle CIR dirigé par un mouvement brownien fractionnaire est défini par la dynamique suivante :

$$dX_t = \gamma(t)(\theta(t) - X_t)dt + \sigma(t)\sqrt{x}dB_t^H$$

Ce modèle est utilisé afin de modéliser l'évolution de taux d'intérêts.

On a : $f(x, t) = \gamma(t)(\theta(t) - x)$, $g(x, t) = \sigma(t)\sqrt{x}$. On a donc $L = \frac{1}{2} \frac{\sigma(t)}{\sqrt{x}} + \frac{\sigma(t)}{\sigma(t)} \frac{-1}{2} x^{-\frac{3}{2}} \theta(t) - \frac{1}{2} x^{-\frac{1}{2}} - Ht^{2H-1} \frac{-\sigma(t)}{4} x^{-\frac{3}{2}}$

Puis on a : $\frac{\partial}{\partial x} g(x, t) L = x^{-2} (\frac{1}{2} \gamma(t) \theta(t) - \frac{\sigma(t)^2}{4} Ht^{2H-1})$

On remarque que pour $\sigma(t) = \sqrt{\frac{2\gamma(t)\theta(t)}{Ht^{2H-1}}}$, $\frac{\partial}{\partial x} g(x, t) L$ s'annule et donc d'après la méthode décrite dans 1.6.1 on peut écrire l'équation différentielle stochastique sous forme linéaire. Nous devons d'abord déterminer $\alpha(t)y$ comme décrit à la section 1.6.1 :

$$\alpha(t)y + \beta(t) = 2x^{\frac{1}{2}} \frac{\partial}{\partial t} \frac{1}{\sigma(t)} + \frac{\gamma(t)}{\sigma(t)} (\theta(t)x^{-\frac{1}{2}} - x^{\frac{1}{2}}) - \frac{1}{2} Ht^{2H-1} \sigma(t)x^{-\frac{1}{2}}$$

On a donc :

$$\alpha(t)y + \beta(t) = x^{\frac{1}{2}} \left(-\frac{\gamma(t)}{\sigma(t)} + 2 \frac{\partial}{\partial t} \frac{1}{\sigma(t)} \right)$$

Or $\beta(t)$ vaut 0 on a donc :

$$\left(\left(\int \alpha(t) dt \right)' + \left(\frac{\gamma(t)}{2} + \sigma(t) \frac{\partial}{\partial t} \frac{1}{\sigma(t)} \right) \right) \int \alpha(t) dt = 0$$

On peut donc égaliser :

$$\int \alpha(t) dt = e^{-\int \frac{\gamma(t)}{2} dt} \sigma(t)^{-1}$$

On obtient finalement :

$$y = 2e^{\int \frac{\gamma(t)}{2} dt} \sqrt{x}$$

On peut donc écrire l'équation différentielle stochastique linéaire :

$$dY_t = e^{\int \frac{\gamma(t)}{2} dt} \sigma(t) dB_t^H$$

On obtient la solution en intégrant :

$$Y_t = \int_0^t e^{\int \frac{\gamma(t)}{2} dt} \sigma(t) dB_t^H + K$$

Enfin, on obtient donc la solution du modèle CIR :

$$X_t = \left(\frac{1}{2} \exp \left(- \int \frac{1}{2} \gamma(t) dt \right) \left(\int_0^t \exp \left(\int \frac{1}{2} \gamma(t) dt \right) \sigma(t) dB_t^H + \sqrt{x(0)} \right) \right)^2$$

On utilise maintenant un schéma numérique de Milstein afin de donner une solution numérique de l'expression plus haut, on a donc :

$$X_{j+1} = X_j + (\gamma(\theta - X_j) - \frac{1}{2}H\sigma^2 t_j^{2H-1})\Delta t + \sigma\sqrt{(X_j)}\Delta B_j^H + \frac{1}{4}\sigma^2(\Delta B_j^H)^2$$

Quelques simulations du modèle CIR fractionnaire ...

Simulation du modèle CIR fractionnaire pour différentes valeurs de H

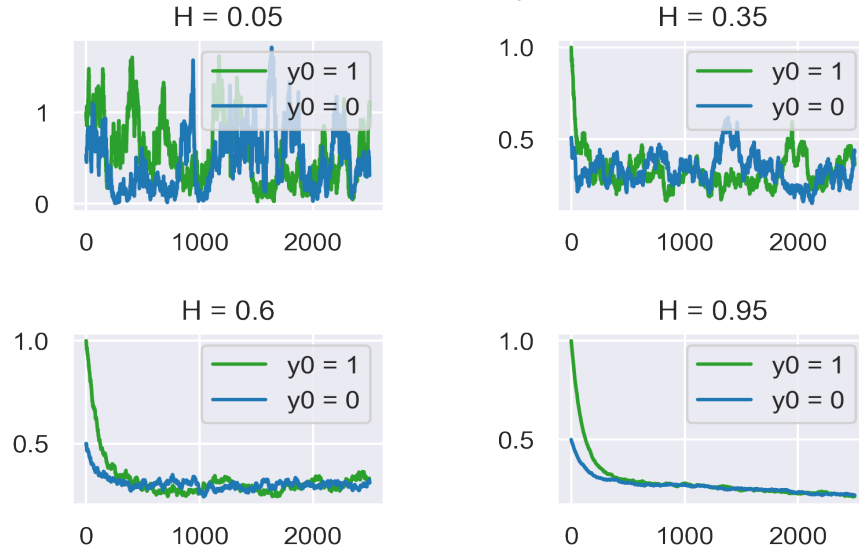


FIGURE 7 – Simulations du modèle CIR fractionnaire pour différentes valeurs de H

1.6.5 Le modèle de Black-Scholes fractionnaire

Le modèle de Black-Scholes existe aussi en version fractionnaire, la partie aléatoire est modélisée par un mouvement brownien fractionnaire. Cependant, ce modèle n'est pas très utilisé en pratique car il écarte l'hypothèse d'absence d'opportunités d'arbitrages

1.6.6 Modélisation d'un actif sous-jacent sous Black-Scholes fractionnaire

La seule différence faite avec le modèle de Black Scholes classique se trouve au niveau du mouvement brownien utilisé, en effet on a :

$$dS_t = \mu S_t + \sigma S_t dB_t^H$$

Afin de simuler le modèle de Black-Scholes fractionnaire nous utiliserons le schéma de Milstein, on a donc :

$$S_{t_{i+1}} = S_{t_i} + rS_{t_i}\Delta t - \sigma^2 S_{t_i} H t_i^{2H-1} \Delta t + \sigma S_{t_i} \Delta B_i^H + \frac{1}{2} \sigma^2 S_{t_i} (\Delta B_i^H)^2$$

Quelques simulations du modèle de Black-Scholes fractionnaire ...

Simulation du modèle de Black-Scholes fractionnaire pour différentes valeurs de H

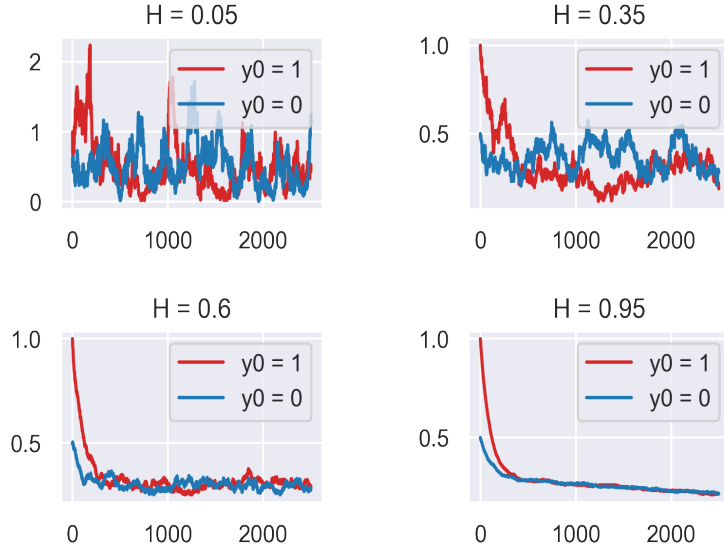


FIGURE 8 – Simulations du modèle de Black-Scholes fractionnaire pour différentes valeurs de H

1.7 Modèle Rough Heston

1.7.1 Modèle de Heston classique

Le Modèle de Heston standard est un modèle à volatilité stochastique très utilisé, il est défini par la dynamique de prix suivante :

$$dS_t = S_t \sqrt{V_t} dW_t$$

De plus, il est régi par un processus de variance décrit par l'équation différentielle suivante :

$$dV_t = \lambda(\theta - V_t)dt + \lambda\nu\sqrt{V_t}dB_t$$

avec $\langle dW_t, dB_t \rangle = \rho dt$, avec $\rho \in [-1; 1]$

W_t et B_t sont deux mouvements browniens standard

1.7.2 Modèle de Heston dirigé par un mouvement brownien fractionnaire

Dans le modèle de Heston fractionnaire, le processus de volatilité est dirigé par un mouvement brownien fractionnaire, comme vu précédemment, selon Gatheral et Rosenbaum la log-volatilité se comporte comme un mouvement brownien fractionnaire de paramètre de Hurst $H \in]0; 1[$.

De plus, cette modélisation (par un mBF) ne modifie en rien les propriétés statistiques de la volatilité. Ainsi l'équation différentielles sont données par :

$$dS_t = S_t \sqrt{V_t} dW_t^H$$

Où : B_t^H et W_t^H sont deux mouvements browniens fractionnaires

La solution donnée pour dV_t [Cours Rosenbaum] est la suivante :

$$V_t = V_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \lambda(\theta - V_s) ds + \frac{\lambda\nu}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \sqrt{V_s} dB_s$$

Selon Alos, Fukasawa et Bayer, de tels modèles permettent de bien reproduire le comportement de la surface de volatilité implicite, en particulier la dépendance de la volatilité par rapport au strike (à la monnaie).

Dans le modèle de Heston classique la fonction caractéristique des log-prix (X_t) est donnée par : $X_t = \log(S_t/S_0)$, en particulier cette équation satisfait :

$$\mathbb{E}[e^{iaX_t}] = \exp(g(a, t) + V_0 h(a, t))$$

Où h est la solution de l'équation de Riccati suivante :

$$\partial_t h = \frac{1}{2}(-a^2 - ia) + \lambda(ia\rho\nu - 1)h(a, s) + \frac{(\lambda\nu)^2}{2}h^2(a, s), \quad h(a, 0) = 0 \quad (1)$$

$$L(t, \lambda, V_t, S_t) = \mathbb{E}[e^{i\lambda \log(S_T)} | F_t]$$

Avec F_t la filtration engendrée par les browniens dirigeants la volatilité et le prix.

L étant une martingale, on a en appliquant la formule d'Itô l'équation suivante (les termes en dt doivent être nuls) :

$$\partial_t L + \lambda(\theta - V)\partial_V L + \frac{1}{2}(\lambda v)^2 V \partial_{VV}^2 L + \frac{1}{2} S^2 V \partial_S^2 L + \rho v \lambda S V \partial_{SV}^2 L = 0$$

1.7.3 Construction du modèle Rough Heston

Nous reprendrons la construction du modèle Rough Heston tel que Rosenbaum l'a construit. On voudrait un modèle tick-by-tick qui prend en compte l'aspect fractionnaire des marchés et qui reproduit les propriétés importantes du modèle de Heston :

- L'effet de levier : La capacité d'amplification de la capacité d'investissement.
- Prendre en compte l'effet d'asymétrie (Kurtosis), de queue lourde.
- La modélisation permet de reproduire la dynamique du volatility smile (Dépendance de la volatilité implicite par rapport au Strike K), comme prévu par le modèle de Black-Scholes on remarque que la volatilité implicite n'est pas constante mais dépendante du strike K

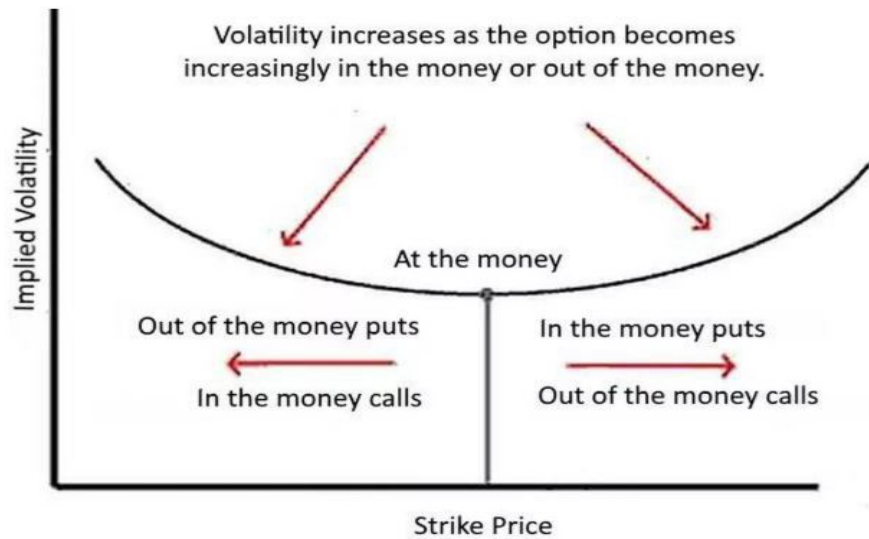


FIGURE 9 – Exemple de volatility smile

En pratique le modèle de Heston, et donc le modèle Rough Heston sont utilisés afin de pricer des options exotiques, mais cette modélisation tend à créer une divergence entre le pricing d'options vanilles (faussés) et la modélisation de ces derniers par un modèle à volatilité constante/déterministe (plus réel). Cependant un problème subsiste concernant le modèle Rough Heston, ce dernier n'est plus Markovien on ne peut donc plus utiliser l'approche classique utilisé dans les modèles à EDS dirigés par un mouvement brownien standard, c'est à dire utiliser la formule d'Itô, nous ne rentrerons pas dans les détails sur ce point.

L'approche qui a été utilisé par El Euch et Rosenbaum consiste à regarder des fonctionnelles de processus de Hawkes qui convergent vers le modèle de Rough Heston, l'idée étant que l'on connaît les fonctions caractéristiques des processus de Hawkes, on peut donc facilement à la limite et donc obtenir des formules pour le modèle Rough Heston. La fonction caractéristique du log-prix dans le modèle Rough Heston montre la même structure que la fonction caractéristique du modèle de Heston classique.

A l'exception que l'équation de Riccati (1) est remplacée par une équation de Riccati fractionnaire où une dérivée fractionnaire est utilisé à la place d'une dérivée classique, on a plus précisément :

$$\mathbb{E}[e^{iaX_t}] = \exp(g_1(a, t) + V_0 g_2(a, t))$$

Avec :

$$g_1(a, t) = \theta \lambda \int_0^t h(a, s) ds, g_2(a, t) = I^{1-\alpha} h(a, t),$$

De plus, $h(a, \cdot)$ est une solution de l'équation de Riccati fractionnaire suivant :

$$D^\alpha h(a, t) = \frac{1}{2}(-a^2 - ia) + \lambda(ia\rho v - 1)h(a, t) + \frac{(\lambda v)^2}{2}h^2(a, t), \quad I^{1-\alpha}h(a, 0) = 0,$$

Avec D^α et $I^{1-\alpha}$ la dérivée fractionnaire et l'opérateur intégrant définit comme suit :

Intégrale fractionnaire Pour $f : \mathbb{R} \rightarrow \mathbb{R}$ appartenant à $L^1(\mathbb{R})$ on peut définir l'opérateur intégrale fractionnaire de Riemann-Liouville d'ordre $\alpha > 0$ par :

$$I^{1-\alpha}f(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x \frac{f(t)}{(x-t)^\alpha} dt$$

Dérivée fractionnaire Dans les mêmes conditions que pour l'intégrale fractionnaire, on peut définir l'opérateur de dérivée fractionnaire comme suit :

$$D^\alpha f(x) = \frac{d}{dx} I^{1-\alpha} f(x)$$

Il convient de remarquer que lorsque $\alpha = 1$, les résultats coïncident avec les résultats du modèle de Heston classique, cependant lorsque $\alpha < 1$ la solution de l'équation de Riccati ne peuvent être établis explicitement, cependant, elles peuvent être facilement numériquement résolue.

Finalement, la loi du processus $(S_t^{t_0}, V_t^{t_0})_{t \geq 0} = (S_{t+t_0}, V_{t+t_0})_{t \geq 0}$ est la loi du modèle Rough Heston dont les dynamiques de volatilités et de prix sont les suivants :

$$dS_t^{t_0} = S_t^{t_0} \sqrt{V_t^{t_0}} dW_t^{t_0}; \quad S_0^{t_0} = S_{t_0}$$

$$V_t^{t_0} = V_{t_0} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \lambda(\theta^{t_0}(s) - V_s^{t_0}) ds + \frac{\lambda\nu}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \sqrt{V_s^{t_0}} dB_s^{t_0}$$

Avec : $(W_t^{t_0}, B_t^{t_0}) = (W_{t_0+t}, B_{t_0+t})$ et θ^{t_0} est un processus F_t -mesurable, dépendant de $(V_u)_{0 \leq u \leq t_0}$.

On peut aussi donner une forme généralisée du modèle Rough Heston :

$$dS_t = S_t \sqrt{V_t} dW_t$$

$$V_t = V_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \lambda(\theta^0(s) - V_s) ds + \frac{\lambda\nu}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \sqrt{V_s} dB_s,$$

Avec $\langle dW_t, dB_t \rangle = \rho dt$, $\alpha \in (1/2, 1)$.

1.7.4 Démonstration de la loi du couple $(S_t^{t_0}, V_t^{t_0})$ conditionnellement à F_t

Comme l'on démontré El Euch et Rosenbaum, il est nécessaire d'utiliser le théorème de Fubini stochastique, l'objectif est de montrer que $I^{1-\alpha}V$ est une semi-martingale pour $t > 0$, posons :

$$(I^{1-\alpha}V)_t = V_0 \int_0^t \frac{s^{-\alpha}}{\Gamma(1-\alpha)} ds + \int_0^t \lambda(\theta^0(s) - V_s) ds + \int_0^t \nu \sqrt{V_s} dB_s$$

On a donc :

$$\frac{1}{\Gamma(1-\alpha)} \int_0^{t+t_0} (t+t_0-u)^{-\alpha} V_u du$$

qui vaut :

$$\frac{1}{\Gamma(1-\alpha)} \int_0^{t_0} (t_0-u)^{-\alpha} V_u du + V_0 \int_{t_0}^{t+t_0} \frac{1}{\Gamma(1-\alpha)} u^{-\alpha} du + \int_{t_0}^{t+t_0} \lambda(\theta^0(u) - V_u) du + \int_{t_0}^{t+t_0} \nu \sqrt{V_u} dB_u$$

En utilisant un changement de variable $t' = t - t_0$, on a :

$$\frac{1}{\Gamma(1-\alpha)} \int_0^{t_0} (t_0-u)^{-\alpha} V_u du + V_0 \int_0^t \frac{1}{\Gamma(1-\alpha)} (t_0+u)^{-\alpha} du + \int_0^t \lambda(\theta^0(u+t_0) - V_u^{t_0}) du + \int_0^t \nu \sqrt{V_u^{t_0}} dB_u^{t_0}$$

Avec $(B_t^{t_0})_{t \geq 0} = (B_{t+t_0} - B_{t_0})$ est un mouvement brownien indépendant de F_{t_0}
De plus, on a :

$$I^{1-\alpha} V_t^{t_0} = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-u)^{-\alpha} V_u^{t_0} du = \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t+t_0} (t+t_0-u)^{-\alpha} V_u du$$

qui vaut :

$$\frac{1}{\Gamma(1-\alpha)} \int_0^{t+t_0} (t+t_0-u)^{-\alpha} V_u^{t_0} du - \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t_0} (t+t_0-u)^{-\alpha} V_u du$$

Et,

$$\frac{1}{\Gamma(1-\alpha)} ((t_0-u)^{-\alpha} - (t+t_0-u)^{-\alpha}) = \frac{\alpha}{\Gamma(1-\alpha)} \int_0^t (t+t_0-u)^{-1-\alpha} dv$$

Puis on dérive :

$$\begin{aligned} I^{1-\alpha} V_t^{t_0} &= \frac{1}{\Gamma(1-\alpha)} \int_0^{t_0} \int_0^t (t_0-u+v)^{-1-\alpha} dv V_u du + \int_0^t (t_0+u)^{-\alpha} du V_0 \\ &\quad + \int_0^t \lambda(\theta^0(u+t_0) - V_u^{t_0}) du + \int_0^t \nu \sqrt{V_u^{t_0}} dB_u^{t_0} \quad (3) \end{aligned}$$

On peut encore réécrire cela comme suit :

$$V_{t_0} \frac{t^{1-\alpha}}{(1-\alpha)\Gamma(1-\alpha)} + \int_0^t \lambda(\theta^{t_0}(u) - V_u^{t_0}) du + \int_0^t \nu \sqrt{V_u^{t_0}} dB_u^{t_0},$$

Où $(\theta^{t_0}(u))_{u \geq 0}$ est une fonction F_{t_0} -mesurable et définit par :

$$\theta^{t_0}(u) = \theta^0(t_0+u) + \frac{\alpha}{\lambda\Gamma(1-\alpha)} \int_0^{t_0} (t_0-v+u)^{-1-\alpha} (V_u - V_{t_0}) dv + \frac{(u+t_0)^{-\alpha}}{\lambda\Gamma(1-\alpha)} (V_0 - V_{t_0})$$

Propriétés de θ^{t_0} θ^{t_0} est continue sur \mathbb{R}_+^+ et. de plus, pour tout $u > 0$ on a :

$$\theta^{t_0}(u) = \theta^0(t_0+u) + \frac{\alpha}{\lambda\Gamma(1-\alpha)} \int_0^{t_0} (t_0-v+u)^{-1-\alpha} V_u dv + \frac{1}{\lambda\Gamma(1-\alpha)} (V_0(u+t_0)^{-\alpha} - V_{t_0} u^{-\alpha})$$

V est un processus non-négatif et θ^0 . Finalement, pour $\epsilon > 0$, V est un processus $\alpha - 1/2 - \epsilon$ Hölder-continue, autrement dit pour chaque $\omega \in \Omega$, et avec $c_\epsilon(\omega)$ tel que pour $x, y \in [0, t_0]$ on a :

$$|V_x - V_y| \leq c_\epsilon(\omega) |x - y|^{\alpha-1/2-\epsilon}$$

Ensuite, par intégration par parties on a, pour tout $u \in (0, t_0]$:

$$\begin{aligned} \left| \int_0^{t_0} (t_0 - v + u)^{-1-\alpha} (V_v - V_{t_0}) dv \right| &\leq c_\epsilon(\omega) \int_0^{t_0} t_0 (t_0 - v + u)^{-1-\alpha} (t_0 - v)^{\alpha-1/2-\epsilon} dv \\ &= c_\epsilon(\omega) u^{-1/2-\epsilon} \int_0^{t_0/u} (x+1)^{-1-\alpha} x^{\alpha-1/2-\epsilon} dx \\ &\leq c_\epsilon(\omega) u^{-1/2-\epsilon} \int_0^\infty (x+1)^{-1-\alpha} x^{\alpha-1/2-\epsilon} dx \end{aligned}$$

Pour finir nous avons :

$$\int_0^t V_s^{t_0} ds = \frac{1}{\Gamma\alpha} \int_0^t 0t(t-s)^{\alpha-1} I^{1-\alpha} V_s^{t_0} ds$$

qui vaut :

$$V_{t_0} t + \frac{1}{\alpha} \int_0^t \int_0^s (s-u)^{\alpha-1} \lambda(\theta^{t_0}(u) - V_u^{t_0}) du ds + \frac{1}{\Gamma(\alpha)} \int_0^t \int_0^s (s-u)^{\alpha-1} \nu \sqrt{V_u^{t_0}} dB_u^{t_0} ds$$

On conclut, en différentiant l'équation au dessus que la dynamique de (S^{t_0}, V^{t_0}) est donnée par :

$$S_t^{t_0} = S_{t_0} \exp \left(\int_0^t \sqrt{V_u^{t_0}} dW_u^{t_0} - \frac{1}{2} \int_0^t V_u^{t_0} du \right)$$

$$V_t^{t_0} = V_{t_0} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-u)^{\alpha-1} \lambda(\theta^{t_0}(u) - V_u^{t_0}) du + \frac{1}{\Gamma(\alpha)} \int_0^t (t-u)^{\alpha-1} \nu \sqrt{V_u^{t_0}} dB_u^{t_0}$$

Où : $(W_t^{t_0})_{t \geq 0} = (W_{t+t_0} - W_{t_0})_{t \leq 0}$ est un mouvement brownien indépendant de F_{t_0} et tel que : $\langle dW_t, dB_t \rangle = \rho$

1.7.5 Schéma numérique de calcul de la fonction caractéristique du log-prix dans le modèle Rough Heston

Comme l'on expliqué El Euch et Rosenbaum, l'objectif est de calculer numériquement la fonction caractéristique, nous avons :

$$D^\alpha h(a, t) = \frac{1}{2}(-a^2 - ia) + \lambda(iap\nu - 1)h(a, t) + \frac{(\lambda\nu)^2}{2}h^2(a, t), \quad I^{1-\alpha}h(a, 0) = 0, \quad (1)$$

Avec :

$$F(a, x) = \frac{1}{2}(-a^2 - ia) + \lambda(iap\nu - 1)x + \frac{(\lambda\nu)^2}{2}x^2, \quad h(a, 0) = 0$$

Bien qu'il existe plusieurs schéma numérique permettant de résoudre numériquement (1), l'approche utilisée est basée sur le fait que (1) implique l'équation de Volterra suivante :

$$h(a, t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} F(a, h(a, s)) ds$$

Puis de développer un schéma numérique basé sur l'équation précédente. En utilisant la méthode fractionnaire Adams, où l'idée est la suivante :

Posons : $g(a, t) = F(a, h(a, t))$, puis discrétisons la fonction h en utilisant un pas Δ tel que ($t_k = k\Delta$), l'estimation de h :

$$h(a, t_{k+1}) = \frac{1}{\Gamma(\alpha)} \int_0^{t_{k+1}} (t_{k+1} - s)^{\alpha-1} g(a, s) ds$$

Est donnée par :

$$\frac{1}{\Gamma(\alpha)} \int_0^{t_{k+1}} (t_{k+1} - s)^{\alpha-1} \hat{g}(a, s) ds$$

Avec :

$$\hat{g}(a, t) = \frac{t_{j+1} - t}{t_{j+1} - t_j} \hat{g}(a, t_j) + \frac{t - t_j}{t_{j+1} - t_j} \hat{g}(a, t_{j+1}), \quad t \in [t_j, t_{j+1}], \quad 0 \leq j \leq k$$

Cela nous amène au schéma suivant :

$$\hat{h}(a, t_{k+1}) = \sum_{0 \leq j \leq k} a_{j,k+1} F(a, \hat{h}(a, t_j)) + a_{k+1,k+1} F(a, \hat{h}(a, t_{k+1}))$$

Où :

$$a_{0,k+1} = \frac{\Delta^\alpha}{\Gamma(\alpha+2)} (k^{\alpha+1} - (k-\alpha)(k+1)^\alpha)$$

$$a_{j,k+1} = \frac{\Delta^\alpha}{\Gamma(\alpha+2)} ((k-j+2)^{\alpha+1} + (k-j)^{\alpha+1} + (k-j)^{\alpha+1} - 2(k-j+1)^{\alpha+1}), \quad 1 \leq j \leq k$$

De plus,

$$a_{k+1,k+1} = \frac{\Delta^\alpha}{\Gamma(\alpha+2)}$$

On calcule ensuite une pre-estimation de $\hat{h}(a, t_{k+1})$ basée sur une somme de Riemann

$$\hat{h}^P(a, t_{k+1}) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \tilde{g}(a, s) ds,$$

Où :

$$\tilde{g}(a, t) = \tilde{g}(a, t_j), \quad t \in [t_j, t_{j+1}), \quad 0 \leq j \leq k$$

Donc on a :

$$\tilde{h}^P(a, t_{k+1}) = \sum_{0 \leq j \leq k} b_{j,k+1} F(a, \hat{h}(a, t_j))$$

Avec :

$$b_{j,k+1} = \frac{\Delta^\alpha}{\Gamma(\alpha+1)} ((k-j+1)^\alpha - (k-j)^\alpha), \quad 0 \leq j \leq k$$

Et finalement on a donc :

$$\hat{h}(a, t_{k+1}) = \sum_{0 \leq j \leq k} a_{j,k+1} F(a, \hat{h}(a, t_j)) + a_{k+1,k+1} F(a, \hat{h}^P(a, t_j)), \quad \hat{h}(a, 0) = 0,$$

En théorie, nous devons avoir les résultats de convergence suivants, pour $t > 0$ et $a \in \mathbb{R}$:

$$\max_{t_j \in [0, t]} |\hat{h}(a, t_j) - h(a, t_j)| = o(\Delta)$$

et, $\epsilon > 0$

$$\max_{t_j \in [\epsilon, t]} |\hat{h}(a, t_j) - h(a, t_j)| = o(\Delta^{2-\alpha})$$

1.8 Le modèle rough SABR

1.8.1 Le modèle SABR classique

Le modèle SABR (stochastic-alpha-bêta-rho) est un modèle à volatilité stochastique utilisé principalement sur les marchés FOREX et les marchés à taux d'intérêt, l'intérêt de ce modèle est que l'on a explicitement une formule fermée pour la volatilité implicite, sa la dynamique est donnée par :

$$dF_t = \sigma_t (F_t)^\beta dW_t$$

$$d\sigma_t = \alpha \sigma_t dB_t$$

Avec f_t : Forward, et $\langle dW_t, dB_t \rangle = \rho dt$,

La formule de volatilité implicite est donnée par :

$$\sigma_B(K, f) = \frac{\alpha \left(1 + \left[\frac{(1-\beta)^2}{24} \frac{\alpha^2}{(fK)^{(1-\beta)}} + \frac{1}{4} \frac{\rho\beta v\alpha}{(fK)^{(1-\beta)/2}} + \frac{2-3\rho^2}{24} v^2 \right] T \right)}{(fK)^{(1-\beta)/2} \left[1 + \frac{(1-\beta)^2}{24} \ln^2 \frac{f}{K} + \frac{(1-\beta)^4}{1920} \ln^4 \frac{f}{K} \right]} \frac{z}{\chi(z)}$$

Avec :

$$z = \frac{v}{\alpha} (fK)^{(1-\beta)/2} \ln \frac{f}{K}$$

$$\chi(z) = \ln \left[\frac{\sqrt{1 - 2\rho z + z^2} + z - \rho}{1 - \rho} \right]$$

1.8.2 Le modèle SABR dirigé par un mouvement brownien fractionnaire

1.9 Le modèle RFSV

Conclusion

A L'approche de Carr et Madan

Carr et Madan propose une approche intéressant afin d'utiliser la transformé de Fourier rapide pour évaluer le prix des options.

En connaissant la fonction caractéristique, on peut, par des formules d'inversion de Fourier obtenir le prix d'un actif, notamment dans le cas que l'on évoquera ici, le prix d'un call en fonction de la fonction caractéristique.

L'idée étant que l'on connaît souvent la fonction caractéristique dans certains modèles, comme le modèle de Heston.

Si on connaît la densité du processus S_t on peut utiliser la formule de transfert pour calculer le prix du Call, ici $L = e^k$ et donc :

$$C_T(k) = e^{-rT} \mathbb{E}^{\mathbb{Q}}[(S_T - L)^+] \quad (4)$$

$$= e^{-rT} \int_k^\infty (e^x - e^k) q(x) dx \quad (5)$$

On ne peut pas exprimer le Call en utilisant le log-strike car $C(k)$ tends vers S_0 quand $k \rightarrow -\infty$ On a donc :

$$C_T(k) = e^{-rT} \int_{-\infty}^\infty (e^x - e^{-\infty}) q(x) dx, \quad (6)$$

$$= e^{-rT} \int_{-\infty}^\infty e^x q(x) dx, \quad (7)$$

$$= e^{-rT} \mathbb{E}^{\mathbb{Q}}[e^{S_t}] \quad (8)$$

On en déduit, en utilisant la propriété des martingales $\mathbb{E}^{\mathbb{Q}}[S_T] = S_0 e^{rT}$ on s'aperçoit que $\lim_{k \rightarrow -\infty} C(k) = S_0$ ne converge pas vers 0. Cependant, $C(k)$ n'est pas L^1 , une transformation de Fourier n'existe pas, cependant en introduisant un facteur exponentielle $e^{\alpha k}$ avec $\alpha > 0$, il est possible $C(k)$ une fonction intégrable.

$$c_T(k) = e^{\alpha k} C_T(k),$$

$c_T(k)$ est une fonction intégrable, car $\int_{-\infty}^\infty |e^{\alpha k} C_T(k)| dk < \infty$, est bien fini, pour α déterminé de manière optimale. La transformée de Fourier de $c(k)$ est donnée par :

$$\begin{aligned} \Psi(u) &= \int_{-\infty}^\infty e^{iuk} c(k) dk, \\ &= \int_{-\infty}^\infty e^{iuk} \int_{-\infty}^\infty e^{\alpha k} e^{rT} (e^x - e^k)^+ q(x) dx dk, \\ &= \int_{-\infty}^\infty e^{iuk} \int_k^\infty e^{\alpha k} e^{-rT} (e^x - e^k) q(x) dx dk, \end{aligned}$$

$$= \int_{-\infty}^{\infty} e^{-rT} q(x) \left(\int_{-\infty}^x (e^x - e^k) e^{iuk} e^{\alpha k} dk \right) dx$$

L'intégrale interne représente le payoff vaut :

$$\begin{aligned} \int_{-\infty}^{\infty} (e^x - e^k) e^{iuk} e^{\alpha k} dk &= e^x \int_{-\infty}^x e^{(\alpha+iu)k} dk - \int_{-\infty}^x e^{(\alpha+iu+1)k} dk, \\ &= \frac{e^x}{\alpha+iu} [e^{(\alpha+iu)k}]_{-\infty}^x - \frac{1}{\alpha+iu+1} [e^{(\alpha+iu+1)k}]_{-\infty}^x \end{aligned}$$

En prenant la limite pour $\lim_{k \rightarrow -\infty} e^{(\alpha+iu)k} = 0$ avec $\alpha > 0$ on a :

$$\left(\frac{e^{(\alpha+1+iu)x}}{\alpha+iu} - \frac{e^{(\alpha+1+iu)x}}{\alpha+iu+1} \right)$$

Ce qui nous amène :

$$\begin{aligned} \Psi(u) &= \int_{-\infty}^{\infty} e^{-rT} q(x) \left(\frac{e^{(\alpha+1+iu)x}}{\alpha+iu} - \frac{e^{(\alpha+1+iu)x}}{\alpha+iu+1} \right) dx, \\ &= e^{-rT} \int_{-\infty}^{\infty} q(x) \frac{e^{(\alpha+1+iu)x}}{(\alpha+iu)(\alpha+1+iu)} dx. \end{aligned}$$

En prenant la transformé de Fourier pour : $\int_{-\infty}^{\infty} q(x) e^{(\alpha+1+iu)x} = \int_{-\infty}^{\infty} q(x) e^{i(u-(\alpha+1))x}$ on a la fonction caractéristique pour le prix sous la probabilité risque neutre $\phi_T(u - (\alpha+1)i)$ finalement on a :

$$\Psi(u) = \frac{e^{-rT} \phi_T(u - (\alpha+1)i)}{\alpha^2 + \alpha - u^2 + i(2\alpha+1)u'}$$

On a donc :

$$\begin{aligned} C_T(k) &= \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{\infty} e^{-iuk} \Psi(u) du, \\ &= \frac{e^{-\alpha k}}{\pi} \int_0^{\infty} \Re[e^{-iuk} \Psi(u)] du \end{aligned}$$

On peut en déduire une formule similaire pour le put, on peut donc répliquer de manière théorique, toutes fonctions en utilisant uniquement des Calls et des Puts (Continuum Calls/Puts). On écrit donc :

$$f(S_T) = f(\kappa) + f'(\kappa)(S - \kappa) + \int_{-\infty}^{\kappa} f''(K)(K - S)^+ dK + \int_{\kappa}^{\infty} f''(K)(S - K)^+ dK$$

Elle est parfois utilisée pour le pricing variance swap sur l'indice VIX.

B Code Python pour simulations et estimations

B.1 Code Python pour l'estimation du paramètre de Hurst H

EstimationParamH

September 23, 2021

```
[22]: #Ce code a été grandement inspiré par le code original
      → de Jim Gatheral ("Volatility is rough"), certaines parties ont même été
      → gardées tel que le sont sur le fichier python original

      →

      → #EstimationParamètresH
import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt
from MFDFA import MFDFA
from MFDFA import fgn
import datetime as dt
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import yfinance as yf
import pandas_datareader.data as pdr
import requests
import zipfile as zi
from sklearn import datasets, linear_model
import scipy.special as scsp
import statsmodels.api as sm
import seaborn as sns; sns.set()
%matplotlib inline
import io as sio

[23]: #Etapes
      → Estimation du paramètre de Hurst H

      → #Estimation du paramètre H pour un seul indice
df = pd.read_csv('oxfordmanrealizedvolatilityindices.csv')
volS = df[df["Symbol"] == ".FCHI"]["rv10"]

[24]: #####
      yf.pdr_override()
      start_date = '01-01-2010'
      end_date = '10-08-2021'
```

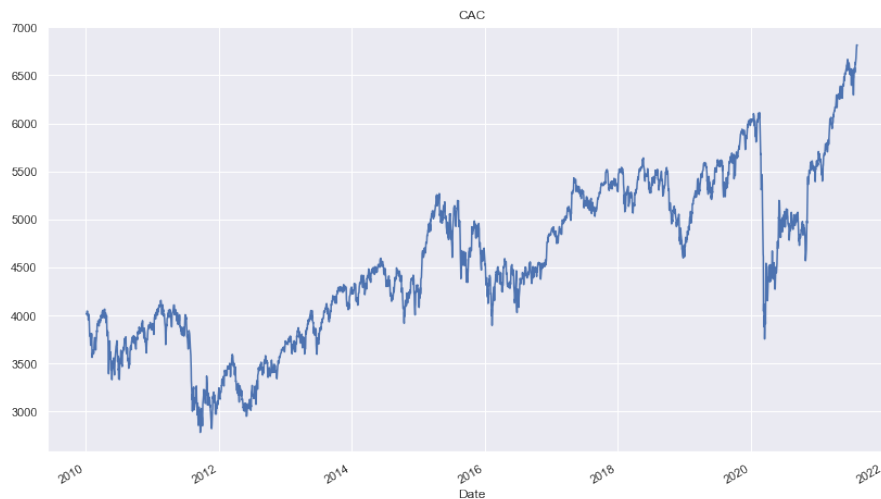


```
def download_data(symbol, source, start_date, end_date):
    start = dt.datetime.strptime(start_date, '%d-%m-%Y')
    end = dt.datetime.strptime(end_date, '%d-%m-%Y')
    df = pdr.get_data_yahoo(symbol, data_source=source, start=start, end=end)
    return df
```

```
[30]: cours = download_data("^FCHI", "yahoo", start_date, end_date)
cours = cours['Adj Close']
cours.plot(title='CAC',figsize=(14, 8))
```

```
#####
[*****100%*****] 1 of 1 completed
```

```
[30]: <AxesSubplot:title={'center':'CAC'}, xlabel='Date'>
```



```
[12]: dfVol = pd.DataFrame()
dfVol['sqrt'] = np.sqrt(volS)
dfVol['log_sqrt'] = np.log(dfVol['sqrt'])

def del_Raw(q, x, df):
    return [np.mean(np.abs(df - df.shift(lag)) ** q) for lag in x]
```

```
[13]: fig = plt.figure(figsize=(8, 8))
ax = fig.add_axes([0.1,0.1,0.75,0.75])
ax.set_xlabel('$\log(\Delta)$')
ax.set_ylabel('$\log m(q,\Delta)$')
```

```

ax.ylim=(-3, -.5)

zeta_q = list()
qVec = np.array([.5, 1, 1.5, 2, 3])
x = np.arange(1, 100)
for q in qVec:
    ax.plot(np.log(x), np.log(del_Raw(q, x, dfVol['log_sqrt'])), 'o')
    model = np.polyfit(np.log(x), np.log(del_Raw(q, x, dfVol['log_sqrt'])), 1)
    ax.plot(np.log(x), np.log(x) * model[0] + model[1])
    zeta_q.append(model[0])

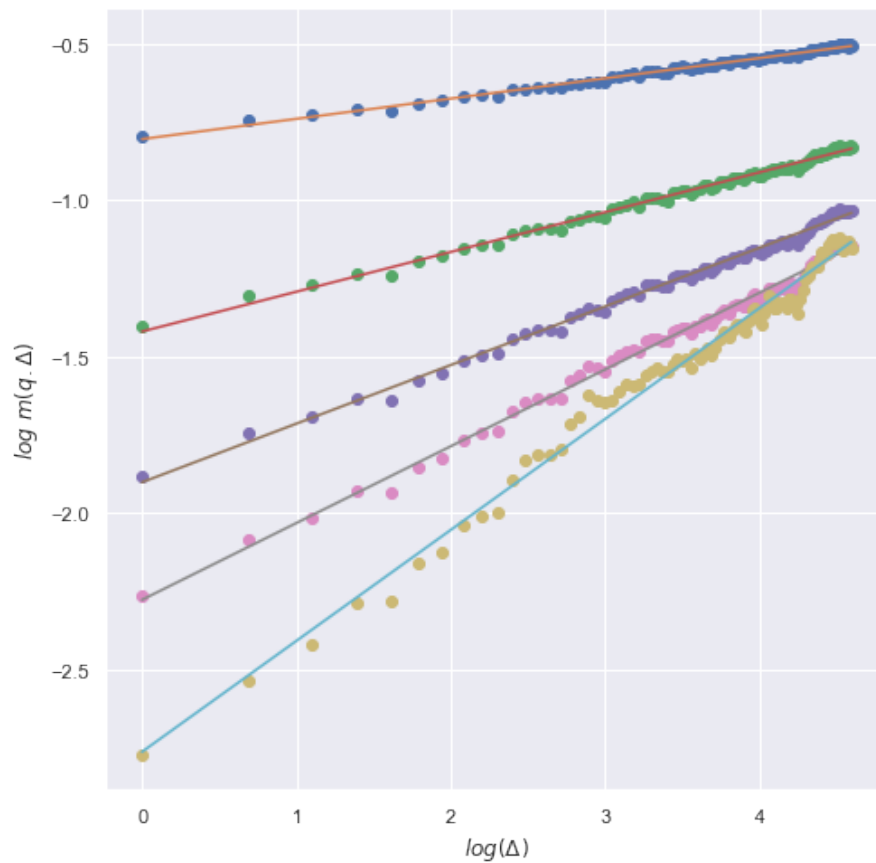
print (zeta_q)
fig.savefig('logM(q,delta)logdelta.png', dpi=300, bbox_inches='tight')

```

```

[0.06465910481822533, 0.12704699654087392, 0.1871885659766945,
0.24513591254023603, 0.3542774580370526]

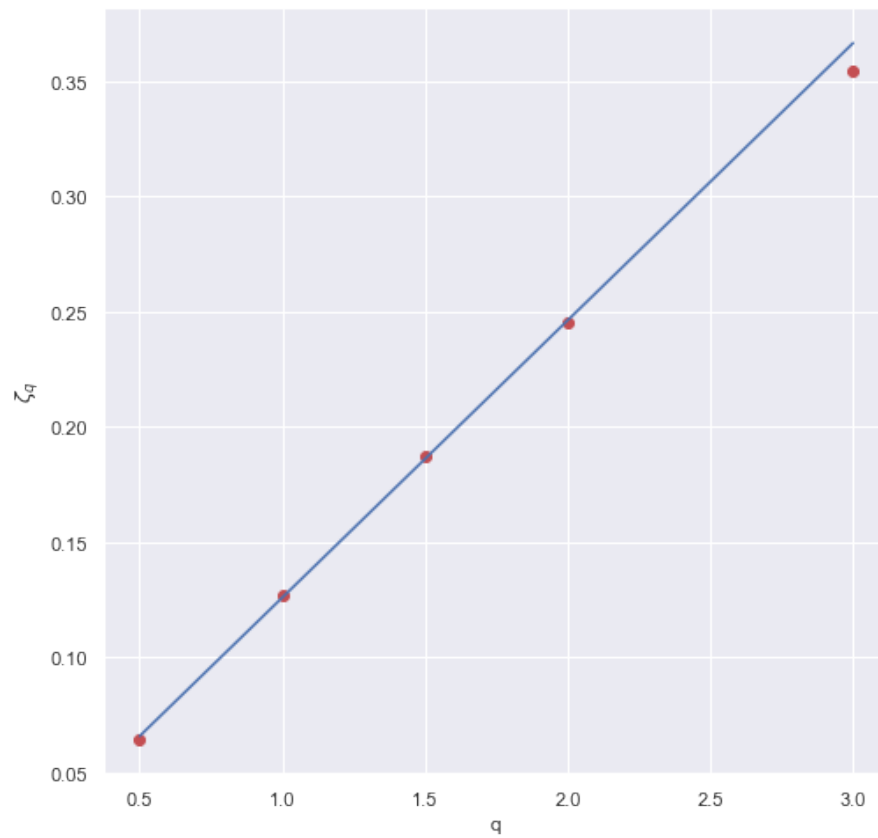
```



```
[14]: plt.figure(figsize=(8,8))
plt.xlabel('q')
plt.ylabel('$\zeta_q$')
plt.plot(qVec, zeta_q, 'or')

line = np.polyfit(qVec[:4], zeta_q[:4],1)
plt.plot(qVec, line[0] * qVec + line[1])
h_est= line[0]
print(h_est)
```

0.12031439852037054



```
[17]: →#Estimation paramètre H pour tous les indices
df = pd.read_csv('oxfordmanrealizedvolatilityindices.csv')
indexes = df["Symbol"].value_counts().index
vol = pd.DataFrame()
for i in range (len(indexes)):
    s = df[df["Symbol"] == indexes[i]]["rv10"]
    vol = pd.concat([vol, s], axis=1)
vol.columns = indexes

def dlsig2(sic, x, pr=False):
    if pr:
        a= np.array([(sig-sig.shift(lag)).dropna() for lag in x])
        a=a ** 2
        print (a.info())
    return [np.mean((sig-sig.shift(lag)).dropna() ** 2) for lag in x]
```

```
[18]: h = list()
nu = list()

for col in vol.columns:
    sig = vol[col]
    sig = np.log(np.sqrt(sig))
    sig = sig.dropna()
    model = np.polyfit(np.log(x), np.log(dlsig2(sig, x)), 1)
    nu.append(np.sqrt(np.exp(model[1])))
    h.append(model[0]/2.)

est = pd.DataFrame({'Indices':vol.columns, 'Estimation du paramètre H': h,
→'Estimation du paramètre nu': nu})
```

/Users/reda/opt/anaconda3/lib/python3.8/site-
packages/pandas/core/arraylike.py:364: RuntimeWarning: divide by zero
encountered in log
result = getattr(ufunc, method)(*inputs, **kwargs)

```
[19]: est
est.dropna()
```

	Indices	Estimation du paramètre H	Estimation du paramètre nu
0	.FCHI	0.122568	0.320464
1	.AEX	0.133145	0.320414
2	.STOXX50E	0.100548	0.397587
3	.BFX	0.125937	0.306290
4	.IBEX	0.112744	0.314685
5	.GDAXI	0.122359	0.331459
6	.AORD	0.071875	0.459212
7	.FTSE	0.109180	0.365529

8	.MXX	0.079091	0.414202
9	.IXIC	0.123072	0.350766
10	.SPX	0.131715	0.381304
11	.RUT	0.110748	0.387228
12	.SSMI	0.149400	0.278083
13	.DJI	0.123347	0.390174
16	.KS11	0.102043	0.319863
17	.BVSP	0.127826	0.333823
18	.HSI	0.084487	0.327221
19	.KSE	0.099285	0.444501
20	.N225	0.105592	0.372272
21	.SSEC	0.105825	0.360319
22	.OSEAX	0.107394	0.369890
23	.GSPTSE	0.110107	0.391804
24	.SMSI	0.109951	0.335748
25	.OMXHPI	0.107753	0.356045
26	.OMXSPI	0.116433	0.354460
27	.OMXC20	0.098222	0.364126
29	.FTMIB	0.114010	0.345525

[]:

B.2 Code Python pour les simulations du mouvement brownien fractionnaire et du processus d'Ornstein-Uhlenbeck

SimulationsMBFetpOUassocie

September 23, 2021

```
[4]: →#SimulationsMBFetProcessusOrnsteinUhlenbeckAssocié
import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt
from MFDFA import MFDFA
from MFDFA import fgn
import datetime as dt
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import yfinance as yf
import pandas_datareader.data as pdr
import requests
import zipfile as zi
from sklearn import datasets, linear_model
import scipy.special as scsp
import statsmodels.api as sm
import seaborn as sns; sns.set()
%matplotlib inline
import io as sio

def choleskyMbf(T, N, H):

    g = lambda k,H: 0.5*(np.abs(k-1)**(2*H) - 2*np.abs(k)**(2*H) + np.
    →abs(k+1)**(2*H))

    m = np.zeros((N,N))
    X = np.zeros(N)
    normalV = np.random.standard_normal(size=N)

    m[0,0] = 1.0
    X[0] = normalV[0]

    m[1,0] = g(1,H)
    m[1,1] = np.sqrt(1 - (m[1,0]**2))
    X[1] = np.sum(m[1,0:2] @ normalV[0:2])
```

```

for i in range(2,N):
    m[i,0] = g(i,H)

    for j in range(1, i):
        m[i,j] = (1/m[j,j])*(g(i-j,H) - (m[i,0:j] @ m[j,0:j]))

    m[i,i] = np.sqrt(1 - np.sum((m[i,0:i]**2)))
    X[i] = m[i,0:i+1] @ normalV[0:i+1]

mBF = np.cumsum(X)*(N**(-H))
return (T**H)*(mBF)

```

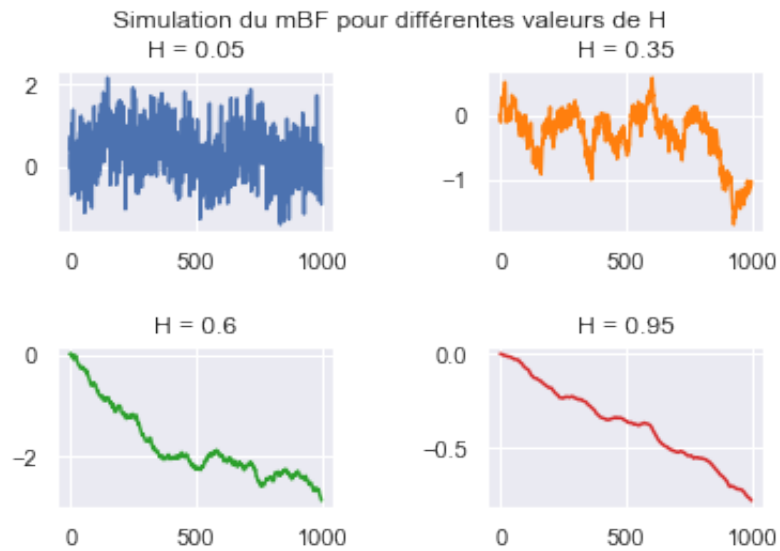
[5]:

```

→#Simulation mouvement brownien fractionnaire
intervale = [0.05, 0.35, 0.6, 0.95]
mBF = []
for i in range(len(intervale)):
    x = choleskyMbf(1, 1000, i)
    mBF.append(x)

fig, axs = plt.subplots(2, 2)
fig.tight_layout(pad=3.0)
fig.suptitle("Simulation du mBF pour différentes valeurs de H")
axs[0, 0].plot(mBF[0])
axs[0, 0].set_title('H = '+str(round(intervale[0], 2)))
axs[0, 1].plot(mBF[1], 'tab:orange')
axs[0, 1].set_title('H = '+str(round(intervale[1], 2)))
axs[1, 0].plot(mBF[2], 'tab:green')
axs[1, 0].set_title('H = '+str(round(intervale[2], 2)))
axs[1, 1].plot(mBF[3], 'tab:red')
axs[1, 1].set_title('H = '+str(round(intervale[3], 2)))
fig.savefig('SimulationsMouvementBrownienFractionnaireH.png', dpi=300)

```



[6]: → *#Simulation processus d'Ornstein-Uhlenbeck fractionnaire*

```
#Paramètres
theta = 0.3
sigma = 0.1
mu = 0.1
t_final = 25
delta_t = 0.01

time = np.arange(0, t_final, delta_t)

listpOU = []
listX = []
listY = []
H = [0.05, 0.35, 0.6, 0.95]
for i in H:
    dBY = (t_final ** i) * fgn(N = time.size, H = i)
    dBX = (t_final ** i) * fgn(N = time.size, H = i)
    x = np.zeros([time.size])
    y = np.zeros([time.size])
    x[0] = 0
    y[0] = 1
    for i in range(1, time.size):
```



```

        x[i] = x[i-1] + theta*(mu - x[i-1])*delta_t + sigma*dBX[i]
        y[i] = y[i-1] + theta*(mu - y[i-1])*delta_t + sigma*dBY[i]
    listX.append(x)
    listY.append(y)

fig, axs = plt.subplots(2, 2)
fig.tight_layout(pad=3.0)
fig.suptitle("Simulation du processus d'Ornstein-Uhlenbeck fractionnaire pour ↵
↵différentes valeurs de H")

axs[0, 0].plot(listY[0], 'tab:green', label='y0 = 1')
axs[0, 0].plot(listX[0], 'tab:blue', label='y0 = 0')
axs[0, 0].set_title('H = '+str(round(intervale[0], 2)))
axs[0, 0].legend(loc="upper right")

axs[0, 1].plot(listY[1], 'tab:green', label='y0 = 1')
axs[0, 1].plot(listX[1], 'tab:blue', label='y0 = 0')
axs[0, 1].set_title('H = '+str(round(intervale[1], 2)))
axs[0, 1].legend(loc="upper right")

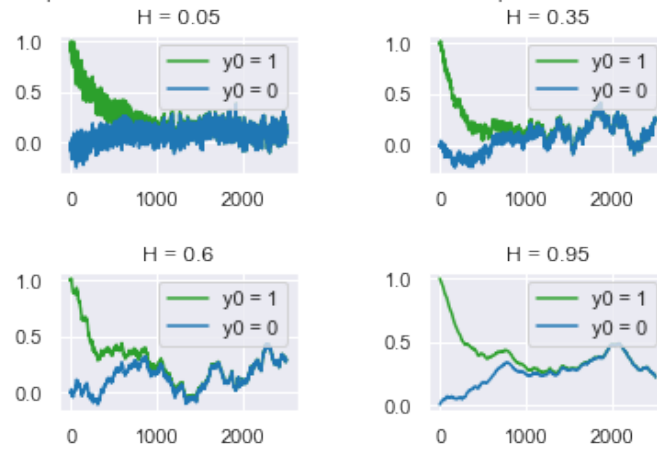
axs[1, 0].plot(listY[2], 'tab:green', label='y0 = 1')
axs[1, 0].plot(listX[2], 'tab:blue', label='y0 = 0')
axs[1, 0].set_title('H = '+str(round(intervale[2], 2)))
axs[1, 0].legend(loc="upper right")

axs[1, 1].plot(listY[3], 'tab:green', label='y0 = 1')
axs[1, 1].plot(listX[3], 'tab:blue', label='y0 = 0')
axs[1, 1].set_title('H = '+str(round(intervale[3], 2)))
axs[1, 1].legend(loc="upper right")

fig.savefig('Simulationsp0-U-H.png', dpi=300, bbox_inches='tight')

```

Simulation du processus d'Ornstein-Uhlenbeck fractionnaire pour différentes valeurs de H



[]:

B.3 Code Python pour les simulations du Modèle de Black-Scholes fractionnaire et du modèle Cox-Ingersoll-Ross

SimulationProcessusCIRetBlackScholes.ipynb

September 23, 2021

```
[106]: ↳#SimulationProcessusCIR
import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt
from MFDFA import MFDFA
from MFDFA import fgn
import datetime as dt
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import yfinance as yf
import pandas_datareader.data as pdr
import requests
import zipfile as zi
from sklearn import datasets, linear_model
import scipy.special as scsp
import statsmodels.api as sm
import seaborn as sns; sns.set()
%matplotlib inline
import io as sio
import os
os.environ['PATH'].split(';')
```

```
[106]: ['/Users/reda/opt/anaconda3/bin:/Users/reda/opt/anaconda3/condabin:/usr/bin:/bin:/usr/sbin:/sbin']
```

```
[85]: def choleskyMbf(T, N, H):

    g = lambda k,H: 0.5*(np.abs(k-1)**(2*H) - 2*np.abs(k)**(2*H) + np.
↳abs(k+1)**(2*H))

    m = np.zeros((N,N))
    X = np.zeros(N)
    normalV = np.random.standard_normal(size=N)

    m[0,0] = 1.0
    X[0] = normalV[0]
```

```

m[1,0] = g(1,H)
m[1,1] = np.sqrt(1 - (m[1,0]**2))
X[i] = np.sum(m[1,0:2] @ normalV[0:2])

for i in range(2,N):
    m[i,0] = g(i,H)

    for j in range(1, i):
        m[i,j] = (1/m[j,j])*(g(i-j,H) - (m[i,0:j] @ m[j,0:j]))

    m[i,i] = np.sqrt(1 - np.sum((m[i,0:i]**2)))
    X[i] = m[i,0:i+1] @ normalV[0:i+1]

mBF = np.cumsum(X)*(N**(-H))
return (T**H)*(mBF)

def CIRFract(r0, gamma, theta, sigma, T, N, H):
    dBX = choleskyMbf(T, N, H)
    delta_t = T/float(N)
    t_i = np.arange(delta_t, T+delta_t, delta_t)
    X = np.zeros(t_i.size)
    X[0] = r0
    for i in range(1, N):
        X[i] = X[i-1] + (gamma*(theta - X[i-1]) - 0.
→5*H*sigma**2*(t_i[i-1]**(2*H-1)))*delta_t + sigma*np.sqrt(X[i-1])*dBX[i-1] +
→0.25*sigma**2*(dBX[i-1])**2
    return(X)

def BSFract(s0, mu, sigma, T, N, H):
    coef = 1
    dBX = choleskyMbf(T, N, H)
    delta_t = T/float(N)
    t_i = np.arange(delta_t, T+delta_t, delta_t)
    X = np.zeros(t_i.size)
    X[0] = s0
    for i in range(1, N):
        X[i] = X[i-1] + r*X[i-1]*delta_t -
→(sigma**2)*X[i-1]*H*t_i[i-1]**(2*H-1) + sigma*X[i-1]*(dBX[i-1]*coef) + 0.
→5*(sigma**2)*X[i-1]*(coef*dBX[i-1])**2
    return(X)

```

[102]:

```

→#Simulation processus d'Ornstein-Uhlenbeck fractionnaire
#Paramètres
theta = 0.3

```

```

sigma = 0.1
mu = 0.1
gamma = 1
t_final = 25
delta_t = 0.01

time = np.arange(delta_t, t_final+delta_t, delta_t)

listX = []
listY = []
listZ = []
listW = []
H = [0.05, 0.35, 0.6, 0.95]
for i in H:
    dBY = (t_final ** i) * fgn(N = time.size, H = i)
    dBX = (t_final ** i) * fgn(N = time.size, H = i)
    dBW = (t_final ** i) * fgn(N = time.size, H = i)
    dBZ = (t_final ** i) * fgn(N = time.size, H = i)
    x = np.zeros([time.size])
    y = np.zeros([time.size])
    z = np.zeros([time.size])
    w = np.zeros([time.size])
    coef = delta_t**i
    x[0] = 1
    y[0] = 1
    z[0] = 0.5
    w[0] = 0.5
    for j in range(1, time.size):
        #Modèle BS frac
        x[j] = x[j-1] + 0.1*x[j-1]*delta_t -
→(sigma**2)*x[j-1]*i*time[j-1]**(2*i-1) + sigma*x[j-1]*(dBX) + 0.
→5*(sigma**2)*x[j-1]*(dBX)**2
        w[j] = w[j-1] + 0.1*w[j-1]*delta_t -
→(sigma**2)*w[j-1]*i*time[j-1]**(2*i-1) + sigma*w[j-1]*(dBW) + 0.
→5*(sigma**2)*x[j-1]*(dBW)**2
        #Modèle CIR frac
        y[j] = y[j-1] + (gamma*(theta - y[j-1]) - 0.
→5*i*sigma**2*(time[j-1]**(2*i-1)))*delta_t + sigma*np.sqrt(y[j-1])*(dBY) + 0.
→25*sigma**2*(dBY)**2
        z[j] = z[j-1] + (gamma*(theta - z[j-1]) - 0.
→5*i*sigma**2*(time[j-1]**(2*i-1)))*delta_t + sigma*np.sqrt(z[j-1])*(dBZ) + 0.
→25*sigma**2*(dBZ)**2
        listY.append(y)
        listZ.append(z)
        listX.append(x)
        listW.append(w)

```

```
[103]: intervalle = H

fig, axs = plt.subplots(2, 2)
fig.tight_layout(pad=3.0)
fig.suptitle("Simulation du modèle CIR fractionnaire pour différentes valeurs de H")

axs[0, 0].plot(listY[0], 'tab:green', label='y0 = 1')
axs[0, 0].plot(listZ[0], 'tab:blue', label='y0 = 0')
axs[0, 0].set_title('H = '+str(round(intervalle[0], 2)))
axs[0, 0].legend(loc="upper right")

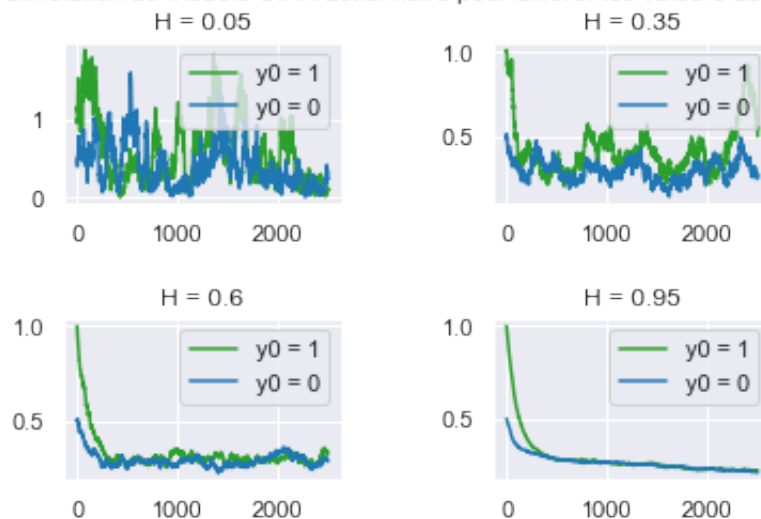
axs[0, 1].plot(listY[1], 'tab:green', label='y0 = 1')
axs[0, 1].plot(listZ[1], 'tab:blue', label='y0 = 0')
axs[0, 1].set_title('H = '+str(round(intervalle[1], 2)))
axs[0, 1].legend(loc="upper right")

axs[1, 0].plot(listY[2], 'tab:green', label='y0 = 1')
axs[1, 0].plot(listZ[2], 'tab:blue', label='y0 = 0')
axs[1, 0].set_title('H = '+str(round(intervalle[2], 2)))
axs[1, 0].legend(loc="upper right")

axs[1, 1].plot(listY[3], 'tab:green', label='y0 = 1')
axs[1, 1].plot(listZ[3], 'tab:blue', label='y0 = 0')
axs[1, 1].set_title('H = '+str(round(intervalle[3], 2)))
axs[1, 1].legend(loc="upper right")

fig.savefig('SimulationsCIRfrac.png', dpi=300, bbox_inches='tight')
```

Simulation du modèle CIR fractionnaire pour différentes valeurs de H



```
[ ]: intervalle = H

fig, axs = plt.subplots(2, 2)
fig.tight_layout(pad=3.0)
fig.suptitle("Simulation du modèle de Black-Scholes fractionnaire pour_
↳différentes valeurs de H")
axs[0, 0].plot(listX[0], 'tab:red', label='y0 = 1')
axs[0, 0].plot(listW[0], 'tab:blue', label='y0 = 0')
axs[0, 0].set_title('H = '+str(round(intervale[0], 2)))
axs[0, 0].legend(loc="upper right")

axs[0, 1].plot(listX[1], 'tab:red', label='y0 = 1')
axs[0, 1].plot(listW[1], 'tab:blue', label='y0 = 0')
axs[0, 1].set_title('H = '+str(round(intervale[1], 2)))
axs[0, 1].legend(loc="upper right")

axs[1, 0].plot(listX[2], 'tab:red', label='y0 = 1')
axs[1, 0].plot(listW[2], 'tab:blue', label='y0 = 0')
axs[1, 0].set_title('H = '+str(round(intervale[2], 2)))
axs[1, 0].legend(loc="upper right")

axs[1, 1].plot(listX[3], 'tab:red', label='y0 = 1')
axs[1, 1].plot(listW[3], 'tab:blue', label='y0 = 0')
axs[1, 1].set_title('H = '+str(round(intervale[3], 2)))
axs[1, 1].legend(loc="upper right")

fig.savefig('SimulationsBSfrac.png', dpi=300, bbox_inches='tight')
```

```
[ ]:
```

C Bibliographie

Références

- [1] Mathieu Rosenbaum, Jim Gatheral, Thibault Jaisson *Volatility is rough*
- [2] Masaaki Fukasawa, Jim Gatheral *A rough SABR formula*
- [3] Mathieu Rosenbaum, Omar El Euch *The characteristic function of rough Heston Models*
- [4] Elisa Alos, David Garcia-Lorite, Aitor Muguruza *On smile properties of volatility derivatives and exotic products : understanding the VIX skew*
- [5] Mathieu Rosenbaum, Paul Jusselin *No-arbitrage implies power-law market impact and rough volatility*
- [6] Fabienne Comte, Eric Renault *LONG MEMORY IN CONTINUOUS-TIME STOCHASTIC VOLATILITY MODELS*
- [7] Gaël Didier *Application du mouvement Brownien fractionnaire à l'évaluation des engagements sur des contrats en unité de compte*
- [8] Ton Dieker *Simulation of Fractional Brownian Motion*
- [9] Jean-François Coeurjolly *Inférence statistique pour les mouvements browniens fractionnaires et multifractionnaires*
- [10] Christian Bender *An Itô formula for generalized functionals of a fractional Brownian motion with arbitrary Hurst parameter*
- [11] Mathieu Rosenbaum *Rough Heston models : Pricing and hedging*
- [12] Peter Carr, Dilip B. Madan *Option Valuation using the fast Fourier Transform*
- [13] Omar El Euch *Quantitative Finance Under Rough Volatility*
- [14] David Nualart *The Malliavin Calculus And Related Topics*
- [15] Mathieu Rosenbaum, Omar El Euch *Perfect hedging in rough Heston models*
- [16] Elisa Alos, Kenichiro Shiraya *Estimating the Hurst parameter from short term volatility swaps : a Malliavin calculus approach*
- [17] Fabrice Douglas Rouah *The SABR model*
- [18] Gazanfer Unal, Ali Dinler *Exact linearization of one dimensional Itô equations driven by fBm : Analytical and numerical solutions*
- [19] Sami El Rahouli *Financial modeling with Volterra processes and applications to options, interest rates and credit risk*
- [20] Marc M. Mpanda, Safari Mukeru, Mmboniseni Mulaudzi *Generalisation of Fractional Cox-Ingersoll-Ross Process*

- [21] Yuliya Mishura, Anton Yurchenko-Tytarenko *Fractional Cox–Ingersoll–Ross process with non-zero «mean»*
- [22] David Nualart, Samy Tindel *A CONSTRUCTION OF THE ROUGH PATH ABOVE FRACTIONAL BROWNIAN MOTION USING VOLTERRA’S REPRESENTATION*
- [23] Christian Bayer, Peter Friz, Jim Gatheral *Pricing under rough volatility*
- [24] Mathieu Garcin *Hurst exponents and delampertized fractional Brownian motions*
- [25] Eyal Neuman, Mathieu Rosenbaum *Fractional Brownian motion with zero Hurst parameter : a rough volatility viewpoint*
- [26] Martin Schmelze *Option Pricing Formulae using Fourier Transform : Theory and Application*