Multivariated Normal Inverse Gaussian Process and Applications to Electricity Future Markets: theory and implementation

Matteo Gardini*

15 June 2019

Abstract

In this document we develop a Multivariated Normal Inverse Gaussian process remaning in Lévy framework and a non-Lévy process with Normal Inverse Gaussian margins. We discuss about theory, implementation and calibration of the model and we fit the model on Power Future Market in Germany and France.

1 Theory

In this costruction we follow what has been proposed in [2][1]

1.1 Multivariate time change

Let $X_1(t)$, $X_2(t)$, $Z(t)$ three indipendent subordinators and consider

$$G(t) = (X_1(t) + \alpha_1 Z(t), X_2(t) + \alpha_2 Z(t)) \quad \alpha_1, \alpha_2 > 0 \tag{1}$$

The process $\boldsymbol{G}$ is a multivariate subordianator and it is sill a Lévy process. Now considere a Brownian motion

$$\boldsymbol{B}(t) = (\mu_1 t + \sigma_1 B_1(t), \mu_2 t + \sigma_2 B_2(t)) \tag{2}$$

where $B_1(t)$ and $B_2(t)$ are **indipendent** standard brownian motion. We can use the multivariated subordinator (1) to time change the process in (2) and define:

$$\mathbf{Y}(t) = \begin{pmatrix} Y_1(t) \\ Y_2(t) \end{pmatrix} = \begin{pmatrix} \mu_1 G_1(t) + \sigma_1 B_1(G_1(t)) \\ \mu_2 G_2(t) + \sigma_2 B_2(G_2(t)) \end{pmatrix} \tag{3}$$

The process defined by (3) is still a Lévy process.

*ERG Spa, Genoa and Mathematical Departement University of Genoa. For comments and suggestions write to gardini@dima.unige.it

[1] We assume that the mutlivariated process are all $\mathbb{R}^2$ processes for ease of notation. The theory can be automatically extended to the general case $\mathbb{R}^d$.

1.2 The general model in Lévy framework

It can be shown that the process in (3) satisfies the following condition:

$$\boldsymbol{Y} \stackrel{d}{=} \boldsymbol{Y}^{X} + \boldsymbol{Y}^{\alpha Z}, \tag{4}$$

where $\boldsymbol{Y}^{X} = \left(B_1\left(X_1\right), B_2\left(X_2\right)\right)^T$ and $\boldsymbol{Y}^{\alpha Z} = \left(\tilde{B}_1\left(\alpha_1 Z\right), \tilde{B}_2\left(\alpha_2 Z\right)\right)$. $\boldsymbol{B}\left(t\right)$ and $\tilde{boldsymbol B}\left(t\right)$ are assumed to be **indipendend**. $\boldsymbol{Y}^{X}$ is time changed with indipendent subordinators $\boldsymbol{X}$ and it has indipendent components. $\boldsymbol{Y}^{\alpha Z}$ is time changed with a unique subordinator Z and therefore has dependent components. $\boldsymbol{Y}^{X}$ and $\boldsymbol{Y}^{\alpha Z}$ are indipendent [1].
The difficult task is to correlate subordinated Brownian motion remaning in Lévy framework.
Consider correlated brownian motion in the $\boldsymbol{Y}^{Z}$ component. So let $\boldsymbol{Y}^{X}$ be as above and let $\boldsymbol{B}^{\rho}$ a multidimensional Brownian motion with drift $\mu_j\alpha_j$, correlation ρ_{12} and diffusion term $\sigma_j\sqrt{\alpha_j}$. Let $\boldsymbol{Y}_{\rho}^{Z}$ be a time changed Brownian motion with a common subordinator $\boldsymbol{Y}_{\rho}^{Z} = \boldsymbol{B}^{\rho}\left(Z\left(t\right)\right)$. Define

$$\boldsymbol{Y}_{\rho} = \boldsymbol{Y}^{X} + \boldsymbol{Y}_{\rho}^{Z} \tag{5}$$

It can be shown that the process in (5) is a Lévy processes and that the marginals have the same law of the ones of $\boldsymbol{Y}$.

It's possible to compute the linear correlation between components of the process $\boldsymbol{Y}_{\rho}$. As shown in [2] it is

$$\rho_{\boldsymbol{Y}_{\rho}} = \frac{\rho\sigma_1\sigma_2\sqrt{\alpha_1\alpha_2}\,\mathbb{E}\left[Z\right] + \mu_1\mu_2\alpha_1\alpha_2 V\left(Z\right)}{\sqrt{V\left(Y_1\right)V\left(Y_2\right)}}$$

1.3 The general model in Non-Lévy framework

There is another way to correlate processes. Consider a multivariate subordinator as (1). If we subordinate a mutivariate brownian motion $\boldsymbol{B}^{\rho}\left(t\right) = \left(B_1\left(t\right), B_2\left(t\right)\right)$ with indipendent components and B_1, B_2 indipentent, we remain in Lévy framework. The problem is that if $B_1\left(t\right)$ and $B_2\left(t\right)$ are correlated and we subordinate them using (1) there is no guarantee that the resulting process is a Lévy process. Indeed, this is an open question but it can be shown that the process $\boldsymbol{B}\left(t\right) = \left(B\left(t\right), B\left(2t\right)\right)$ is not a Lévy process. So we guess that $\boldsymbol{B}\left(t\right) = \boldsymbol{B}^{\rho}\left(\boldsymbol{G}\left(t\right)\right)$ is not a Lévy process. Anyway we can use it to model the market.

Let $\boldsymbol{B}^{\rho}\left(t\right)$ a bivariate brownian motion and let $\boldsymbol{G}\left(t\right)$ a multivariate subordinator as in (1). We obtain the process:

$$\mathbf{Y}\left(t\right) = \left(\begin{array}{c} Y_1\left(t\right) \\ Y_2\left(t\right) \end{array}\right) = \left(\begin{array}{c} \mu_1 G_1\left(t\right) + \sigma_1 B_1^{\rho}\left(G_1\left(t\right)\right) \\ \mu_2 G_2\left(t\right) + \sigma_2 B_2^{\rho}\left(G_2\left(t\right)\right) \end{array}\right) \tag{6}$$

If G_i is an Inverse Gaussian process it then the margins of $\boldsymbol{G}\left(t\right)$ are Normal Inverse Gaussian.

1.4 Risk-Neutral modeling

Once we defined the process $\boldsymbol{Y}(t)$ we can model the value of the underlying F_t^i as:

$$F_t^i = F_0^i e^{rt+\omega_i t+Y_i(t)} \tag{7}$$

where r is the risk-free, Y_i is the stochastic process and ω_i is the drift corrector in order to obtain risk-neutral valuation.
If Y_i is a Normal Inverse Gaussian the characteristic function is

$$\Phi_T = \exp\left(\log(S_0) + (r+\omega)\, iTu + T\delta\left(\sqrt{\alpha^2-\beta^2} - \sqrt{\alpha^2-(\beta+iu)^2}\right)\right) \tag{8}$$

where

$$\omega = \delta\left(\sqrt{\alpha^2-(\beta^2+1)} - \sqrt{\alpha^2-\beta^2}\right) \tag{9}$$

2 Pricing Algorithms

In this section we present two methods to price options. The FFT method we present can be used to price univariate options and it's extreamly usefull in calibration process.
The Monte Carlo algorithm that we use to price basket options or spread options. Indeed in a multivariate framework the easier way to price is by using a Monte Carlo algorithms.

2.1 FFT Algorithm

We discussed about FFT algorithm in a previous document. Here we simply report the main results. The Call value can be obtained inverting the Fourier Transform via FFT algorithm computing:

$$C_T(k) = e^{-\alpha k}\frac{1}{\pi}\int_0^\infty e^{-ivk}\frac{e^{-rT}\Phi_T(v-(\alpha+1)i)}{\alpha^2+\alpha-v^2+i(2\alpha+1)v}dv$$

where Φ_T has the form reported in (8).

This algorithm is extremly fast and can be use during the model calibration procedure.

2.2 Monte Carlo Algorithm

The following algorithm can ben used to simulate a correlated multivariate Lévy process of the type presented in Section 7.3.
Here we need a little bit of shrewdness to guarantee that the sum of two process is still one of the same type. Let's look at the details. Consider, as proposed in [2] a multivariate subordinator of the form

$$\mathbf{G}(t) = \begin{pmatrix} G_1(t) \\ G_2(t) \end{pmatrix} = \begin{pmatrix} X_1(t) + \alpha_1 Z(t) \\ X_2(t) + \alpha_2 Z(t) \end{pmatrix} \tag{10}$$

where X_1, X_2 and Z are indipendent subordinators. What we need are conditions that guarante us that the resulting process $\boldsymbol{G}$ is still a subordinator of the same type. To buid a Normal Inverse Gaussian process we proceed as follows. First of all set $X_i \sim IG\left(1 - a\gamma_i, \frac{b}{\gamma_i}\right)$ and $Z \sim IG\,(a, b)$. Moreover we need the following restrictions on parameters:

$$
\begin{aligned}
& b > 0, \\
& 0 < a < \frac{1}{\gamma_i} \quad i = 1, 2
\end{aligned}
$$

If we set $\alpha_i = \gamma_i^2$ it easy to verify (using characteristic functions) that $G_i = X_i + \gamma_i^2 Z \sim IG\left(1, \frac{b}{\gamma_i}\right)$, so $G_i\,(t) \sim IG\left(t, \frac{b}{\gamma_i}\right)$. Now we need some conditions to obtain NIG margins for the process $\boldsymbol{Y}\,(t)$.
From the univariate case we know that a $NIG\,(\alpha, \beta, \delta)$ can be obtained setting:

$$
Y\,(t) = \beta\delta^2 G\,(t) + \delta B\,(G\,(t))
$$

where $B\,(t)$ is a standard brownian motion and G is a Inverse Gaussian process with parameters $a = 1$ and $b = \delta\sqrt{\alpha^2 - \beta^2}$. So, in multivariated case, we have to chose the parameter b_j of the subordinator G_j such that:

$$
b_j = \frac{b}{\gamma_j} = \delta_j \sqrt{\alpha_j^2 - \beta_j^2}
$$

In the end we have that the process

$$
\mathbf{Y}\,(t) = \begin{pmatrix} Y_1\,(t) \\ Y_2\,(t) \end{pmatrix} = \begin{pmatrix} \beta_1\delta_1^2 G_1\,(t) + \delta_j B_1\,(G_1\,(t)) \\ \beta_2\delta_2^2 G_2\,(t) + \delta_j B_2\,(G_2\,(t)) \end{pmatrix} \tag{11}
$$

has $NIG\,(\alpha_i, \beta_i, \delta_i)$ margins.
Remembering that $\boldsymbol{Y}\,(t)$ has the same margins of $\boldsymbol{Y}_\rho$ in (5) we can simulate the process presented in setting:

$$
\mathbf{Y}\,(t) = \begin{pmatrix} Y_1\,(t)^X + Y_{1\rho}^Z\,(t) \\ Y_2\,(t)^X + Y_{2\rho}^Z\,(t) \end{pmatrix} \tag{12}
$$

using the Algorithm 1.

3 Calibration Issues in Normal Inverse Gaussian Case

We follow the same procedure that we follow in the α-Variance Gamma Process: we first calibrate the marginals parameters on Plain Vanilla options and the we fit the correlation matrix based on forward prices history. From calibration of the marginals we get $\alpha_1, \alpha_2, \beta_1, \beta_2, \delta_1, \delta_2$. Fitting the correlation matrix we get a, b and ρ.
For the fitting of marginal parameters we use non linear least squares and the FFT method to compute options prices. This means that we have to solve

Algorithm 1 Correlated Normal Inverse Gaussian Process Simulation

Set $b_i = \delta_i \sqrt{\alpha_i^2 - \beta_i^2}$
Set $\gamma_i = \frac{b}{b_i}$
Simulate IG processes $X_i\left(t\right) \sim IG\left(1 - \frac{a}{\gamma_i}, \frac{b}{\gamma_1}\right)$
Simulare IG process $Z\left(t\right) \sim IG\left(a, b\right)$.
Set: $dt^{(1)}_{n\Delta t} = X^1_{n\Delta t}\left(t\right) - X^1_{(n-1)\Delta t}\left(t\right)$
Set: $dt^{(2)}_{n\Delta t} = X^2_{n\Delta t}\left(t\right) - X^2_{(n-1)\Delta t}\left(t\right)$
Set: $dt^{Z}_{n\Delta t} = Z_{n\Delta t}\left(t\right) - Z_{(n-1)\Delta t}\left(t\right)$
Generate ν_1, ν_2 indipendent from $\mathcal{N}\left(0, 1\right)$
Generate ν_1^ρ, ν_2^ρ from $\mathcal{N}\left(0, 1\right)$ with correlation ρ.
Set $W^{(1)}_{n\Delta t} = W^{(1)}_{(n-1)\Delta t} + \sqrt{dt^{(1)}_{n\Delta t}}\nu_1$
Set $W^{(2)}_{n\Delta t} = W^{(2)}_{(n-1)\Delta t} + \sqrt{dt^{(1)}_{n\Delta t}}\nu_2$
Set $W^{(1)\rho}_{n\Delta t} = W^{(1)\rho}_{(n-1)\Delta t} + \sqrt{dt^{Z}_{n\Delta t}}\nu_1^\rho$
Set $W^{(2)\rho}_{n\Delta t} = W^{(2)\rho}_{(n-1)\Delta t} + \sqrt{dt^{Z}_{n\Delta t}}\nu_2^\rho$
Set $Y_i^{nc}\left(t\right) = \beta_i \delta_i^2 X_i\left(t\right) + \delta_i W^{(i)}\left(t\right)$
Set $Y_i^{\rho}\left(t\right) = \beta_i \delta_i^2 \gamma_i^2 Z\left(t\right) + \gamma_i \delta_i W^{(i)}\left(t\right)$
return $Y_i\left(t\right) = Y_i^{nc}\left(t\right) + Y_i^{\rho}\left(t\right)$

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} \left(C_i^\theta\left(K, T\right) - C_i\right)^2 \tag{13}$$

where C_i^θ are the model European Call prices and C_i are the market prices. To fit the correlation matrix we use the Generalized Method of Moments applied to the following quantities:

$$\begin{aligned}
\rho_Y &= \frac{\beta_1 \delta_1^2 \beta_2 \delta_2^2 \gamma_1^2 \gamma_2^2 \frac{a}{b^2} + \rho \delta_1 \delta_2 \gamma_1 \gamma_2 a}{\sqrt{\delta_1^2 \gamma_1 + \frac{\beta_1^2 \delta_1^4 \gamma_1^3}{b^2}} \sqrt{\delta_2^2 \gamma_2 + \frac{\beta_2^2 \delta_2^4 \gamma_2^3}{b^2}}} \\
\mathbb{E}\left[Y_1 Y_2\right] &= \beta_1 \beta_2 \delta_1^2 \delta_2^2 \gamma_1^2 \gamma_2^2 \frac{a}{b^3} + \delta_1 \delta_2 \gamma_1 \gamma_2 \rho \frac{a}{b} \\
cov\left(Y_1, Y_2\right) &= \beta_1 \delta_1^2 \beta_2 \delta_2^2 \gamma_1^2 \gamma_2^2 \frac{a}{b^2} + \rho \delta_1 \delta_2 \gamma_1 \gamma_2 a
\end{aligned}$$

4 Numerical Results

For our test we use data from EEX Market. We choose the 12th November 2018 as Settlement Date and we decide to price options on Calendar 2020 in Germany and France. For correlation fitting we select the time series of forward Calendar 2020 prices from the 25th April 2017 and 12th Novembre 2018 for both countries. At settlement date 12th November 2018 we observe the values in Table 1

Now we fit the margins on option prices and we get the following parameters Fitting the the correlation we obtain the parameters shown in Table 3

Parameter	Germany	France
F_0	53.31	58.25
r	0.01	0.01
K	$[28, \cdots, 62]$	$[35.5, \cdots, 66]$
T	31st December 2018	31st December 2018

Table 1: Market values on 12th November 2018

Parameter	Germany	France
α	19.33	50.15
β	2.91	20.58
δ	2.07	3.92

Table 2: Marginal parameters calibration

Parameter	Estimated Value
a	5.41
b	5.43
ρ	0.79
$\rho_{\boldsymbol{Y}_\rho}$	0.2566
$\rho_{\boldsymbol{Y}_\rho}^{max}$	0.4523

Table 3: Marginal parameters calibration

where ρ is the correlation between correlated brownian motion, $\rho_{\boldsymbol{Y}_\rho}$ is the correlation between log-returns of German and France forward prices and $\rho_{\boldsymbol{Y}_\rho}^{max}$ is the maximum correlation between forward prices that can be captured by the model.

Let's look clooser to the fitting. In Figure 1 we see that the France and German option market is well fitted by the marginal NIG processes.
Anyway, looking at the correlated path in Figure we see that the model does not fit properly the log-returns correlation. This is not a surprise if we look at the value of $\rho_{\boldsymbol{Y}_\rho}^{max} = 0.4523$. In Energy forward market we have that, generally, the correlation between log-returns is higher, about $0.9 - 0.95$ and for this reason the model may not be appropriate for pricing spread options. In Figure (2) is reported the spread between France and Germany distribution with maturity of one year for model presented in Section and Section . We see that, even if the Non-Lévy model creates a less wide distribution the value of the spread is unreal. The maximum historical spread between France and Germay forward price for our dataset is $9.04\,[EUR/MWh]$.

In Figure 3 real forward paths are compared with simulated forward path in multivariated Lévy model of Section 7.3. In Figure 4 real forward paths are compared with simulated forward path in multivariated Non-Lévy model of Section 7.4.
In Figure 5 we look at the scatter plot of the log-returns Germany and France

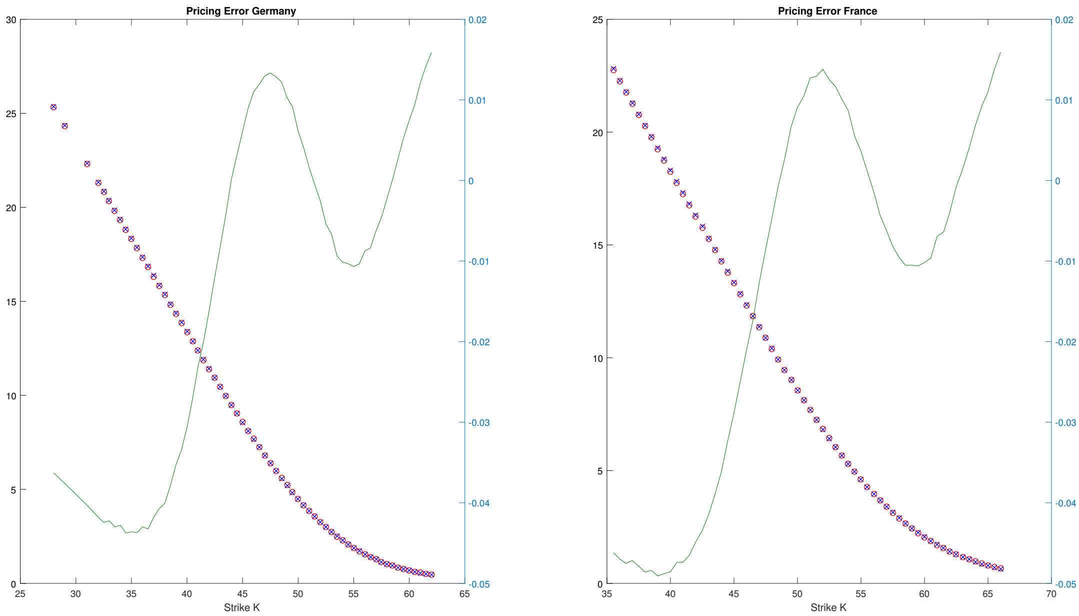

Figure 1: Market fitting of Call values on Forward for German and France market.

forward prices compared to the real log-returns. Again we notice that the strong dependence between the two markets is not well captured by our models.

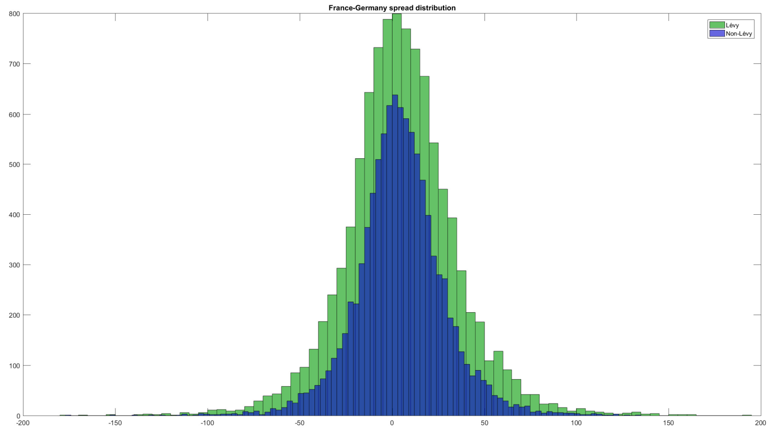

Figure 2: Spread Forward France-Germany distribution maturity one year.

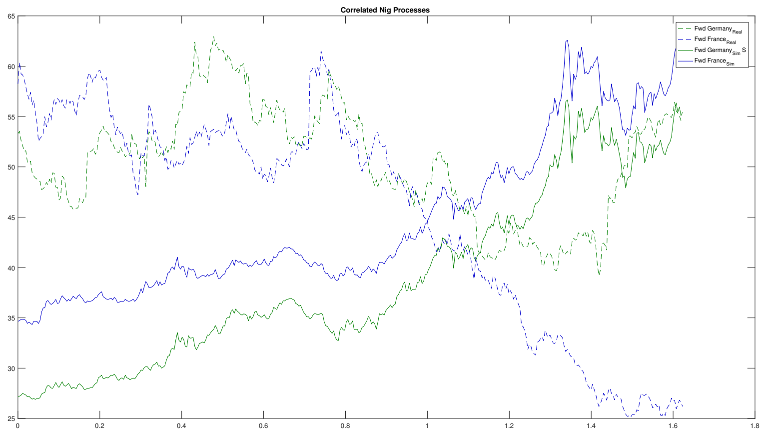

Figure 3: Lévy model paths (dotted) compared to real paths (continuos)

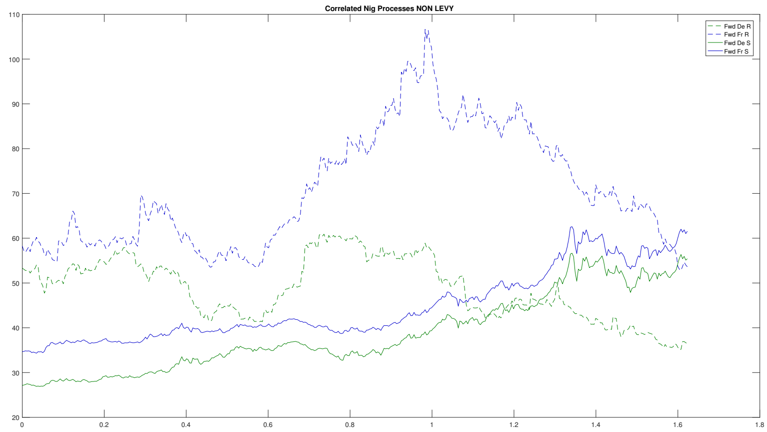

Figure 4: Non-Lévy model paths (dotted) compared to real paths (continuos)

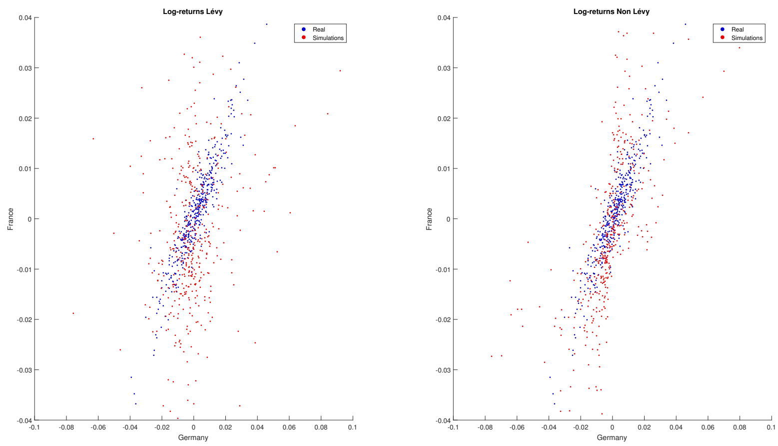

Figure 5: Log-returns scatter plots for Lévy and Non-Lévy models compared to the log returns of real forward prices.

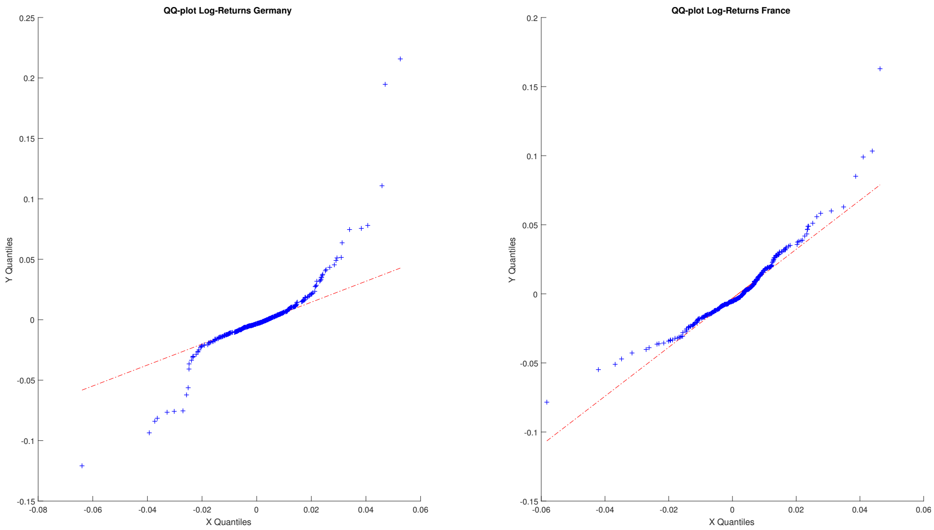

Figure 6: QQ-plot of log-returns of real forward prices vs simulated forward prices

5 Conclusion and further research

6 Open Questions and Observations

6.1 Sull'adeguatezza del modello NIG univariato

I prezzi degli strumenti plain vanilla sono fittati bene dal modello NIG come mostra la Figura 1. Tuttavia, paragonando le distribuzione dei log rendimenti dei prezzi forward reali e simulati in Figura 6, sembrebbe che la distribuzione storica dei log rendimenti non venga poi ben catturata dal modello. Questo potrebbe essere dovuto al fatto che le opzioni quotate su EEX vengano "prezzate" tramite un modello che **non** è NIG e quindi il fitting dei parametri non è buono? Supponiamo, come in realtà è, che EEX prezzi le opzioni usando una qualche versione di Black-Scholes e supponiamo che le i log-rendimenti dei forward seguano un NIG. Teniamo a mente che potrebbe essere un mercato poco liquido. A questo punto io "indovino" il modello ma fittando tramite prezzi che non sono calcolati tramite NIG ottengo dei parametri per il NIG "distorti". E' probabile che il problema consista in questo? Per risolverlo si può pensare di fare una calibrazione sui log-rendimenti storici forwarda e poi cambiare misura tramite trasformata di Esscher in modo da porsi sotto la misura risk-neutral? E' giusto supporre che il mio modello NIG fittato sui dati storici a cui poi applico la trasformata di Esscher non mi replichi i prezzi di mercato che trovo su EEX?

6.2 Sulla correlazione

Sembrerebbe che il modello considerato non riesca a replicare la correlazione dei prodotti forward del mercato elettrico europeo. I log-rendimenti dei forward risultano essere correlati con un valore di $\rho \simeq 0.9 - 0.95$ correlazioni che non riescono ad essere raggiunte dal modello. Come si può fare per ovviare a questa inconvenienza? Uso un subordinatore comune a tutti e due i processi? Oppure cambio modello e cerco di sviluppare una sorta di "modello cointegrato NIG"? Questo modello non va bene per prezzare le opzioni forward sullo spread perchè non replica bene le correlazioni e quindi la distribuzione dello spread è troppo "larga" come si vede in Figura 2.

6.3 Sul modello non Lévy

Sembrerebbe che subordinando due moti browniani correlati con due differenti subordinatori si ottenga un processo che non è più Lévy, sebbene le marginali lo siano. Questo è intuibile visto che il processo $\boldsymbol{B} = (B_1(t), B_1(2t))$. Tuttavia non sembra facile da dimostrare questa cosa nel caso generale. Il fatto che si esca dal framework Lévy sembrerebbe spiegare perchè Luciano-Semeraro adottino la scomposizione $\boldsymbol{Y}_\rho = \boldsymbol{Y}^X + \boldsymbol{Y}^Z_\rho$. Può valere la pena tentare la dimostrazione di questo fatto?

6.4 Applicabilità del modello

Sebbene tale modello non sia buono per simulare i prezzi forward del mercato elettrico europeo potrebbe essere utilizzato per simulare i prezzi delle azioni di aziende dello stesso settore. Ad esempio potrebbe essere usato per simulare l'andamento di aziende del settore energy (A2A, Enel, ERG,...) oppure del settore petrolifero (BP, Eni,...). Infatti questi mercati presentano una minore correlazione dei log-rendimenti ma sicuramente esiste una dipendenda tra l'andamento di questi titoli. Se il settore energetico è in crisi le aziende andaranno più o meno bene ma in qualche modo risentiranno sicuramente della crisi del settore in cui operano.

7 Matlab Codes

In this section we provide the main *MATLAB* codes we used to simulate correlated processes defined in Section 7.3 and in Section 7.4

7.1 Inverse Gaussian variables simulation

This is the code to generate random variables distributed as a $IG(a, b)$ with the following parametrization:

$$f(x; a, b) = \frac{a}{\sqrt{2\pi}} \exp(ab) \, x^{-\frac{3}{2}} \exp\left(-\frac{1}{2}\left(a^2 x^{-1} + bx\right)\right)$$

```
function igVariables = IGrandEfficient(a,b,nSim,nDates)
% genera nSim variabili aleatorie Inverse Gaussian con parametri ...
    a e b
% Genera una matrice di dimensione nSim x nDates di variabili ...
    indipendenti
% distribuite secondo una Gamma
%
% MG 03maggio2019

% inizializzazione
igVariables = zeros(nSim*nDates,1);

v = randn(nSim*nDates,1);

y = v.^2;
x = (a/b) + 0.5.*y./(b^2) + 0.5.*sqrt(4*a*b.*y + y.^2)./b^2;
u = rand(nSim*nDates,1);

idx = u ≤ a./(a + x.*b);

igVariables(idx) = x(idx);
igVariables(¬idx) = a^2./(b^2.*x(¬idx));

% Rendile nel formato matrice con un reshape
igVariables = reshape(igVariables,nSim,nDates);


end
```

7.2 Inverse Gaussian process simulation

This codes simulates Inverse Gaussian process $X(t)$ with density:

$$f(x; a, b) = \frac{at}{\sqrt{2\pi}} \exp(atb) \, x^{-\frac{3}{2}} \exp\left(-\frac{1}{2}\left(a^2 t^2 x^{-1} + bx\right)\right)$$

```
function X = IGProcessSim(a,b,dt,nSim,nDates)
% Genera le simulazioni di un processo Inverse Gaussian con ...
    parametri a e
% b, intervallo temporale dt,nSim simulazioni per nDates date
%
% MG 04maggio2019
```

```

% genera le variabili casuali distribuite come una Inverse Gaussian
igVariables = IGrandEfficient(a*dt,b,nSim,nDates-1);

X = zeros(nSim,nDates);

% Crea il processo
X(:,2:end) = cumsum(igVariables,2);


end
```

7.3 Correlated NIG processes in Lévy framework

This codes generates Normal Inverse Gaussian process in Lévy framework as presented in Section .

```
function [XNIG1,XNIG2,gamma1,gamma2] = ...
    NIGProcessSimHighCorrelated(alpha1,beta1,delta1,...
    alpha2,beta2,delta2,a,b,rho,dt,nSim,nDates)
% function XNIG = ...
    NIGProcessSimCorrelated(alpha,beta,delta,dt,nSim,nDates)
% Simula due NIG correlati
%
%
%
% dt: intervallo temporale
% nDates: numero di date
% nSim: numero di simulazioni

% Simula il primo subordinatore
% Condizione di NIG

b1 = delta1.*sqrt(alpha1^2 - beta1^2);
b2 = delta2.*sqrt(alpha2^2 - beta2^2);

% Gamma
gamma1 = b/b1;
gamma2 = b/b2;


fprintf('Assertion 1 a-1/gamma1: %.2f\n',a-1/gamma1);
fprintf('Assertion 2 a-1/gamma2: %.2f\n',a-1/gamma2);

% Verifica delle condizioni di positivity
assert(a<1/gamma1);
assert(a<1/gamma2);

X1 = IGProcessSim(1-a*gamma1,b/gamma1,dt,nSim,nDates);
X2 = IGProcessSim(1-a*gamma2,b/gamma2,dt,nSim,nDates);

Z = IGProcessSim(a,b,dt,nSim,nDates);

% Trova gli incrementi
deltaT1 = diff(X1,[],2);
deltaT2 = diff(X2,[],2);
deltaTZ = diff(Z,[],2);

%% Simula la parte Browniana Indipendente
```

```
nu1 = randn(nSim,nDates-1);
nu2 = randn(nSim,nDates-1);

%% Simula la parte Brownian Dipendente
[nu1c,nu2c] = SimulaNormaliStandardCorrelate(nSim,nDates-1,rho);

% Inizializza
W1 = zeros(nSim,nDates);
W2 = zeros(nSim,nDates);

W1c = zeros(nSim,nDates);
W2c = zeros(nSim,nDates);

% Scrivi il browniano per il caso indipendente
W1(:,2:end) = cumsum(sqrt(deltaT1).*nu1,2);
W2(:,2:end) = cumsum(sqrt(deltaT2).*nu2,2);

% Scrivi il Browniano per il caso correlato
W1c(:,2:end) = cumsum(sqrt(deltaTZ).*nu1c,2);
W2c(:,2:end) = cumsum(sqrt(deltaTZ).*nu2c,2);

% Crea i nig non correlati
Y1 = beta1*delta1^2.*X1 + delta1*W1;
Y2 = beta2*delta2^2.*X2 + delta2*W2;

% Crea i nig correlati
Y1c = beta1*delta1^2*gamma1^2.*Z + gamma1*delta1*W1c;
Y2c = beta2*delta2^2*gamma2^2.*Z + gamma2*delta2*W2c;


% Crea il processo NIG
XNIG1 = Y1 + Y1c;
XNIG2 = Y2 + Y2c;


end
```

7.4 Correlated NIG processes in Non-Lévy framework

This codes generates Normal Inverse Gaussian process in a non Lévy framework as presented in Section .

```
function [XNIG1,XNIG2] = ...
    NIGProcessSimCorrNonLEVY(alpha1,beta1,delta1,...
    alpha2,beta2,delta2,rho,...
    dt,nSim,nDates)
% Simula un processo Normal Inverse Gaussian con parametri ...
    alpha, beta,
% delta.
% rho: correlazione tra le componenti browniane
% dt: intervallo temporale
% nDates: numero di date
% nSim: numero di simulazioni

% Simula il processo IG
a = 1;
b = delta1*sqrt(alpha1^2-beta1^2);
XIG1 = IGProcessSim(a,b,dt,nSim,nDates);
b = delta2*sqrt(alpha2^2-beta2^2);
```

```
XIG2 = IGProcessSim(a,b,dt,nSim,nDates);


% Trova gli incrementi
deltaT1 = diff(XIG1,[],2);
deltaT2 = diff(XIG2,[],2);


% Simula la parte Browniana
[nu1,nu2] = SimulaNormaliStandardCorrelate(nSim,nDates-1,rho);

% Inizializza
W1 = zeros(nSim,nDates);
W2 = zeros(nSim,nDates);


% Scrivi il browniano
W1(:,2:end) = cumsum(sqrt(deltaT1).*nu1,2);
W2(:,2:end) = cumsum(sqrt(deltaT2).*nu2,2);

% Crea il processo NIG
XNIG1 = beta1*delta1^2.*XIG1 + delta1*W1;
XNIG2 = beta2*delta2^2.*XIG2 + delta2*W2;



end
```

Appendices

A Monte Carlo Algorithm

In this section we verify that the Algorithm 1 has the same distrbution as $Y(t)$ were defined as $Y(t)^X + Y(t)^{\alpha Z}$.

Let:

$$\begin{aligned}
Y &= \mu G(t) + \sigma B(G(t)) \\
Y^X &= \mu X(t) + \sigma B(X(t)) \\
Y^{\alpha Z} &= \mu \alpha Z(t) + \sigma \sqrt{\alpha} \tilde{B}(Z(t)) \\
G(t) &= X(t) + \alpha Z(t)
\end{aligned}$$

whit X, Z, B and $\tilde{B}$ indipendent. We assume that $Y(t)^X$ and $Y(t)^{\alpha Z}$ are indipendent but this can be proven.
Now we compute the characteristic funcion of Y and $Y^X + Y^{\alpha Z}$ (we omit the index t).

$$\begin{aligned}
\mathbb{E}\left[e^{iuY}\right] &= \mathbb{E}\left[e^{iu(\mu G + \sigma B(G))}\right] = \mathbb{E}\left[\mathbb{E}\left[e^{iu\left(\mu g + \sigma\sqrt{g}W\right)} \middle| G = g\right]\right] = \mathbb{E}\left[e^{iu\mu G - \frac{1}{2}\sigma^2 G u^2}\right] \\
&= \mathbb{E}\left[e^{i\left(u\mu + \frac{1}{2}\sigma^2 i u^2\right)G}\right] = \mathbb{E}\left[e^{itG}\right] \\
&= \mathbb{E}\left[e^{it(X+\alpha Z)}\right] \overset{ind.}{=} \mathbb{E}\left[e^{itX}\right]\mathbb{E}\left[e^{it\alpha Z}\right] = \Phi_X(t)\,\Phi_Z(\alpha Z)
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left[e^{iu\left(Y^X + Y^{\alpha Z}\right)}\right] &\overset{ind.}{=} \mathbb{E}\left[e^{iuY^X}\right]\mathbb{E}\left[e^{iuY^{\alpha Z}}\right] = \mathbb{E}\left[e^{iu\mu X + \sigma B(X)}\right]\mathbb{E}\left[e^{iu\mu\alpha Z + \sigma\sqrt{\alpha}\tilde{B}(Z)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[e^{iu\mu x + \sigma\sqrt{x}W} \middle| X = x\right]\right]\mathbb{E}\left[\mathbb{E}\left[e^{iu\mu\alpha z + \sigma\sqrt{\alpha z}W} | Z = z\right]\right] \\
&= \mathbb{E}\left[e^{iu\mu X - \frac{1}{2}\sigma^2 X u^2}\right]\mathbb{E}\left[e^{iu\mu\alpha Z - \frac{1}{2}\sigma^2\alpha Z u^2}\right] \\
&= \mathbb{E}\left[e^{itX}\right]\mathbb{E}\left[e^{i\alpha t Z}\right] = \Phi_X(t)\,\Phi_Z(\alpha Z)
\end{aligned}$$

where we set $t = \mu u + \frac{1}{2}\sigma^2 i u^2$. We can conclude that $Y \overset{d}{=} Y^X + Y^{\alpha Z}$. ■

References

[1] E. Luciano and P. Semeraro. A generalized normal mean variance mixture for return processes in finance. *International Journal of Theoretical and Applied Finance*, 2010.

[2] E. Luciano and P. Semeraro. Multivariate time changes for lévy asset models: Characterization and calibration. *Journal of Computational and Applied Mathematics*, 223:1937–1953, 2010.