Multivariate assed models in Lévy framework and applications to power energy forward market

Matteo Gardini[*]

September 5, 2019

Abstract

In this document we apply the approach prosed in [1] to model the interaction between forwrd power markets in Germany and France. In this document we focus on Normal Inverse Gaussian process.

1 The Model

The model is the one proposed in [1]. For semplicity we consider the 2-dimensional case. The theory can be extendet to the general d-dimensional case. Let $Z\left(t\right), Y_1\left(t\right), Y_2\left(t\right)$ Lévy processes and consider:

$$\boldsymbol{X}\left(t\right) = \left(X_1\left(t\right), X_2\left(t\right)\right) = \left(Y_1\left(t\right) + a_1 Z\left(t\right), Y_2\left(t\right) + a_2 Z\left(t\right)\right)$$

It can be proven that the resulting process $\boldsymbol{X}\left(t\right)$ is a Lévy process. You can note that the distribution of $\boldsymbol{X}\left(t\right)$ is determinated by the distribution of $\boldsymbol{Y}\left(t\right), Z\left(t\right)$. A very useful results for calibraton is the following: it can be proved that:

$$\rho_{12} = \frac{a_1 a_2 Var\left[Z\left(1\right)\right]}{\sqrt{Var\left[X_1\left(t\right)\right]}\sqrt{Var\left[X_2\left(t\right)\right]}} \tag{1}$$

The key idea of the process is to impose some convolution restrictions such that the process $\boldsymbol{Y}\left(t\right) + Z\left(t\right)$ has the same distribution of $\boldsymbol{X}\left(t\right)$. This is very important for the calibration of our model because we can fit the marginal parameters calibrating using plain vanilla options and then we can fit the "common" parameters using the correlation matrix.
So we chose the distributions of $\boldsymbol{X}\left(t\right), \boldsymbol{Y}\left(t\right), Z\left(t\right)$ from the same distribution and then we impose that:

$$\phi_{X_j}\left(t\right) = \phi_{Y_j}\left(t\right) + \phi_Z\left(t\right) \tag{2}$$

where ϕ is the characteristi exponent of the process.
Taking a tempered stable subordinator $G\left(t\right)$ with $\alpha \in [0, 1)$ and characteristic function:

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

$$\phi_G = \frac{\alpha-1}{\alpha k}\left[\left(1-\frac{iuk}{1-\alpha}\right)^{\alpha}-1\right] \tag{3}$$

Now we take subordinated brownian motion $Y_1(t), Y_2(t), Z(t)$ indipendent and of the same family with:

- Y_1 with drift β_1 and volatility γ_1 subordinator params ν_1
- Y_2 with drift β_2 and volatility γ_2 subordinator params ν_2
- Z with drift β_Z and volatility γ_Z subordinator params ν_Z

the $\boldsymbol{X}(t)$ is a subordinate brownian motion of the same family of Y_1, Y_2, Z with drift $\boldsymbol{\theta} = (\theta_1, \theta_2)$ and volatility $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)$ if convolutions conditions (2) are satisfied. If we use (3) and (2) then we have the following:

$$\begin{cases} k_i\theta_i = \nu_Z a_i \beta_Z & j = 1,2 \\ k_i\sigma_i^2 = \nu_Z a_i^2 \gamma_Z^2 & j = 1,2 \end{cases} \tag{4}$$

and consequently:

$$\theta_j = \beta_j + a_j\beta_Z \quad \sigma_j^2 = \gamma_j^2 + a_j^2\gamma_Z^2 \quad k_j = \nu_j\nu_Z/(\nu_j+\nu_Z) \quad j = 1,2 \tag{5}$$

1.1 Modeling

Once we got $\boldsymbol{X}(t)$ we can model each Forward as:

$$S_i(t) = S_i(0)\exp\{X_1(t) + \omega_i t + rt\}$$

where r is the risk-free rate and ω_i is the drift correction that guarantees that discounted processes are martingales. This is obtained setting the characteristic exponent valued at $-i$ equal to zero.
In case of NIG Process it can be easly shown that:

$$\omega_i = -\frac{1}{k_i}\left(1-\sqrt{1-2\theta k_i - \sigma_i^2 k_i}\right)$$

2 Structure Dependence Analysis

It can be shown that, in case of Variance Gamma or Normal Inverse Gaussian process, the correlation coefficient between the two processes is given by:

$$\rho = \frac{a_1a_2\left(\gamma_Z^2 + \beta_Z^2\nu_Z^2\right)}{\sqrt{\sigma_1^2+\theta_1^2k_1}\sqrt{\sigma_2^2+\theta_2^2k_2}} \tag{6}$$

and it is subject to the following conditions in (4) and (5).
Since seems very hard sto study analytically the possibile values of correlation we try in a numerical way. First we generate randomly $\sigma_1, \sigma_2, k_1, k_2, \theta_1, \theta_2$. Then we compute bounds for $\nu_z > \max(k_1, k_2)$ and we generate ν_z, γ_z, β_z. We compute a_1 and a_2 in the following way:

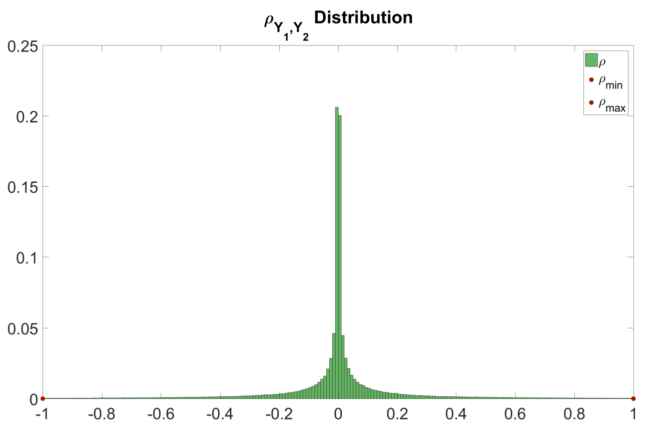


Figure 1: Possible correlations captured by the model. In red the maximum extreme value captured.

$$a_i = \frac{\sigma_i^2 \beta_z}{\theta_i \gamma_z^2}.$$

Then we check the condition:

$$a_i^2 \gamma_z^2 \le \sigma_i^2 \quad i = 1, 2, \tag{7}$$

and if (7) is satisfied we can compute correlation using (6). It can happen that $\rho \notin [-1, 1]$: in this case we drop the value and we generate a different set of parameters. Bounds for parameters are shown in Table 1.

Parameter	**Lowerbound**	**Uppebound**
σ_i $i = 1, 2$	0.01	1
k_i $i = 1, 2$	0.01	2
θ_i $i = 1, 2$	-1.6	1.6
β_z	-5.88	5.88
θ_z	0	0.5

Table 1: Parameters bounds for correlation study

With this set of parameters performing $N_{scen} = 5 \ldots 10^4$ simulatios we get the correlation distribution shown in Figure 1. We notice that extreme values of correlation can be replicated by the model with a proper choice of parameters $a_1, a_2, \gamma_z, \beta_z \nu_z$ given the set $\sigma_1, \sigma_2, k_1, k_2, \theta_1, \theta_2$.

3 Monte Carlo Algorithm for Pricing and Calibration

In this section we propose the Monte Carlo algorithm to simulate the process proposed in the previous section. Moreover, we schetch the algorithm we used to fit the correlation matrix.

3.1 Monte Carlo algorithm for NIG process

The proposed algorithm is valid for the Normal Inverse Gaussian case but can be easly developed to simulate a Variance Gamma process.
The fist step of the algoritm is to simulate a Inverse Gaussian process. There are different parametrization for a Inverse Gaussian density: we use the following one.

$$p(x) = \sqrt{\frac{\lambda}{2\pi x^3}} e^{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}} \mathbb{I}_{x\geq 0}$$

The first step is to simulate an Inverse Gaussian variable: the Algorithm 1 is presented in [2].

Algorithm 1 Inverse Gaussian Simulation

```
1: Generate a random variable $N \sim \mathcal{N}(0,1)$
2: Set $Y = N^2$
3: $X_1 = \mu + \frac{\mu^2 Y}{2\lambda} - \frac{\mu}{2\lambda}\sqrt{4\mu\lambda Y + \mu^2 Y^2}$
4: Generate a uniform $U \sim U[0,1]$
5: if $U \leq \frac{\mu}{X_1+\mu}$ then
6:     return $X_1$
7: else
8:     return $\frac{\mu^2}{X_1}$
```

Now we are ready to simulate a Normal Inverse Gaussian process. Remember that the Normal Inverse Gaussian is obtained subordinating a Brownian motion with drift θ and volatility σ with a Inverse Gaussian process. The Algorithm 2 can be use for this purpose [2].

Algorithm 2 Normal Inverse Gaussian Process

```
.
1: Simulation of $X(t_1), \dots, X(t_n)$ for for fixed time $t_1 \dots, t_n$.
2: Use Algorithm 1 to simulate $n$ indipendente IG variables $\Delta S_1, \dots, \Delta S_n$ with
   parameters $\lambda_i = \frac{(t_i - t_{i-1})^2}{k}$ and $\mu_i = t_i - t_{i-1}$ with $t_0 = 0$.
3: $n$ i.i.d. $\mathcal{N}(0,1)$ random variables $N_1, \dots, N_n$.
4: Set $\Delta X_i = \sigma N_i \sqrt{\Delta S_i} + \theta \Delta S_i$
5: Set the discretize path: $X(t_i) = \sum_{k=1}^{i} \Delta X_i$
```

The two algorithms proposed can be used to simulate the two processes $\boldsymbol{Y}(t)$ and $Z(t)$ that can be combined in the resulting processed $\boldsymbol{X}(t)$. Under convolution conditions (2) we are sure that, if $Y_i(t)$ and $Z(t)$ are two indipendent Normal Inverse Gaussian processes the resulti process $X_i(t)$ is still a Normal Inverse Gaussian process. The Algorithm 3 allow us to simulate the process $\boldsymbol{X}(t)$.

Algorithm 3 Exponential correlated Normal Inverse Gaussian Process

```
1: Use Algorithm 2 to simulate three NIG processes with parameters $Y_1(\beta_1,\gamma_1,\nu_1)$, $Y_2(\beta_2,\gamma_2,\nu_1)$ and $Z(\beta_Z,\gamma_Z,\nu_Z)$.
2: Set $X_i = Y_i + a_i Z, \quad i = 1,2.$
3: Compute $\omega_i = -\frac{1}{k_i}\left(1-\sqrt{1-2\theta k_i - \sigma_i^2 k_i}\right), \quad i = 1,2..$
4: Set $S_i(t) = \exp\{X_i(t) + rt + \omega_i t\}, \quad i = 1,2..$
```

3.2 Fitting parameters

For calibration we follow our usual two-steps procedure. Given the plain vanilla options prices we fit the margins of $\boldsymbol{X}(t)$ getting the parameters θ_i, β_i, k_i $i = 1,2$. Given this parameters we have to fit the correlation between $(X_1(t), X_2(t))$, ρ. It can be shown [1] that the correlation is:

$$\rho = \frac{a_1 a_2\left(\gamma_Z^2 + \beta_Z^2 \nu_Z^2\right)}{\sqrt{\sigma_1^2 + \theta_1^2 k_1}\sqrt{\sigma_2^2 + \theta_2^2 k_2}} \tag{8}$$

From this, given $\sigma_i, \theta_i, k_i \quad i = 1,2$, we can select $(a_1, a_2, \gamma_Z, \beta_Z, \nu_Z)$ that match historical forward correlation and use (5) to obtain the missing parameters β_i, γ_i, ν_i $i = 1,2$.

So the fitting procedure can be devided, as usual, in two steps. For the fitting of marginal parameters we use non linear least squares and the FFT method to compute options prices and we fit the set of parameters that produce the same prices we observe in the market. This means that we have to solve convolution equations 5.

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n}\left(C_i^{\theta}(K,T) - C_i\right)^2 \tag{9}$$

where C_i^{θ} are the model European Call prices and C_i are the market prices. θ is $\theta = (\sigma_1, \theta_1, k_1, \sigma_2, \theta_2, k_2)$.

To fit the correlation matrix we use the Generalized Method of Moments applied to the following quantity:

$$\rho = \frac{a_1 a_2\left(\gamma_Z^2 + \beta_Z^2 \nu_Z^2\right)}{\sqrt{\sigma_1^2 + \theta_1^2 k_1}\sqrt{\sigma_2^2 + \theta_2^2 k_2}}$$

and subject to the non linear costraints:

$$\begin{aligned} &\sigma_i^2 - a_i^2\gamma_Z^2 \geq 0 && i = 1,2 \\ &\nu_Z > k_i && i = 1,2 \end{aligned}$$

that are required to guarantee that γ_i, ν_i are positive. γ_i, ν_i can be simply derived from

Once we obtained the complete set of parameters we can use Monte Carlo Algorithm to simulate forward paths use them to price more complex contracts such as American Option, Spread Options and so on.

4 Numerical Results

In this section we summerize the numerical results obtained with our model. As stated before the calibration is divided in two steps:

- We calibrate the parameters of the margins.
- We fit the covariance structure.

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 2

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 2: Market values on 12th November 2018

4.1 Margins Calibration

Now we fit the margins on option prices and we get the following parameters

Parameter	**Germany**	**France**
θ	0.314	1.760
σ	0.329	0.293
k	0.025	0.006

Table 3: Marginal parameters calibration

To check the goodness of estimatetd parameters we use the estimated parameters to re-price the Vanilla European options quoted on the market. The results is shown in Figure 2: we see that the error is very small so we can assume that the margins parameters have been well calibrated.

4.2 Correlation Matrix Fitting

Fitting the the correlation and using the convolution equation (5) we obtain the parameters shown in Table 4

We observe that the numerical correlation simulation beetween log-returns is higher that in previous model. For this reason, this model seems to be better to model power forward prices. In Figure 3 we plot the distribution of correlation and in Figure 4 we have the scatter plot of simulated log-returns against the real log-returns.

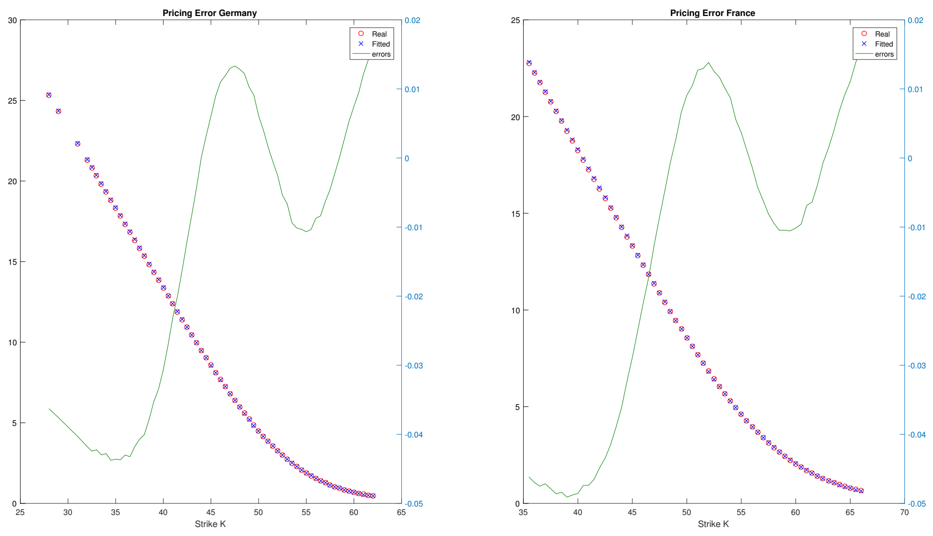


Figure 2: Market fitting of Call values on Forward for German and France market. We used the calibrated parameters to reprice Options quoted on the market. The re-pricing errori is very small.

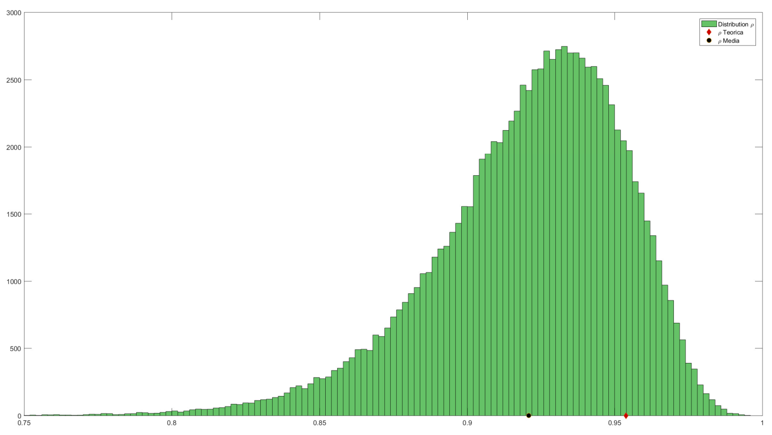


Figure 3: Log-returns correlation distribution.

Parameter	Estimated Value
a_1	5.41
a_2	5.43
β_Z	1.709
γ_Z	2.250
ν_Z	0.1
β_1	0.065
β_2	1.1537
γ_1	0.001
γ_2	0.000
ν_1	0.034
ν_2	0.006
$\rho_{Teorica}$	0.95
$\rho_{meanNumerical}$	0.92

Table 4: Marginal parameters calibration.

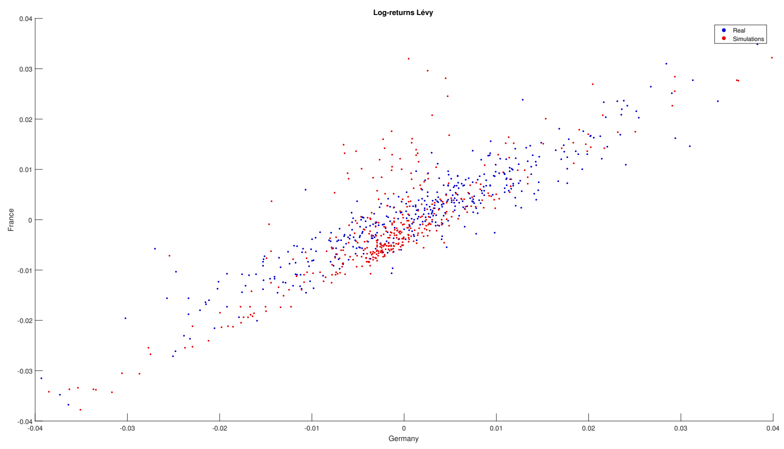


Figure 4: Log-returns scatter plot. In blue real data and in red simulation data.

Due to the higher correlation fitted, the distribution of the spread between the two commodities is tighter that in the previous models as can be seen in Figure 5.
Now we use the set of simulated price via Monte Carlo method to revaluate the quoted European Options. We do this test to check the stabilitity and the correctness of Monte Carlo method. The result is shown in Figure 6. We see that the prices obtained using Monte Carlo method is very close to the ones observed in the market. We can conclude then both calibration and Monte Carlo method are correct. Using Monte Carlo method we can price more complex derivatives. In Figure 7 are displayed some simulated paths for the Normal Inverse Gaussian model.

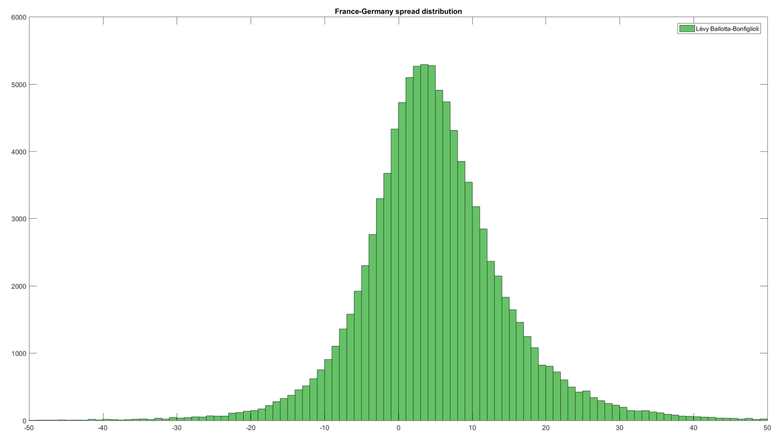


Figure 5: Spread Distribution.

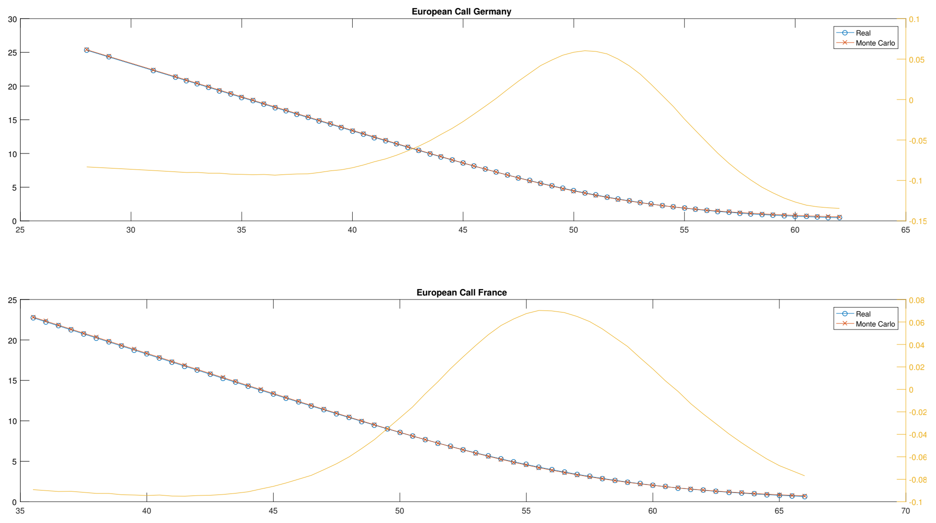
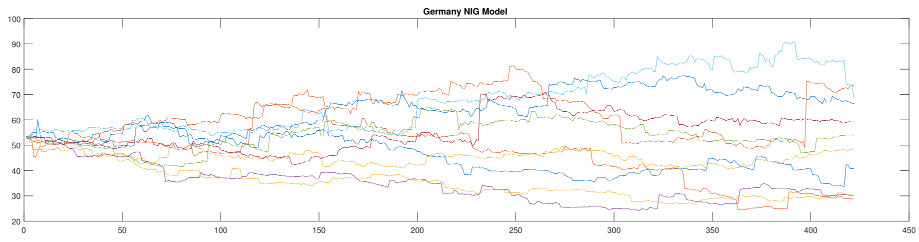


Figure 6: Repricing of the market using the Monte Carlo method.



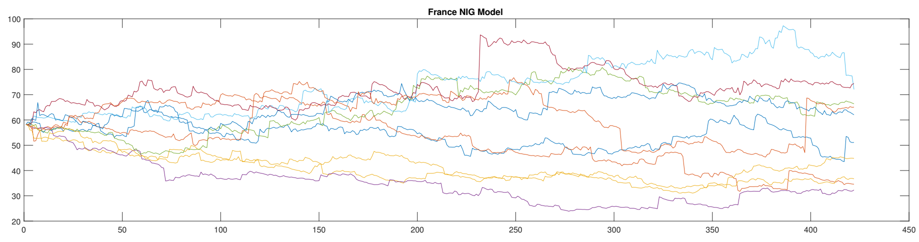


Figure 7: Some plots of simulated paths in Normal Inverse Gaussian model.

5 Conclusions

6 Comments and Open Questions

- Il fitting della matrice di correlazione è veramente tedioso. E' un problema di minimizzazione non lineare vincolato con vincoli non lineari. Esiste un modo migliore per stimare tali parametri? Potrebbe essere utile introdurre metodi di statistica Bayesiana per la stima dei parametri? Essendo che i parametri da stimare sono relativamente pochi, un algoritmo genetico per il fitting della matrice di correlazione potrebbe fare al caso nostro?

- Le correlazioni di questo modello sono nettamente più alte di quelle ottenibili seguendo l'approccio Semeraro e Semeraro-Luciano. Nonostante ciò la distributione numerica assume una forma "inusuale". Come è possibile interpretare questa la distribuzione della correlazione numerica? Come mai la media della distribuzione non coincide con la distribuzone teorica calcolata tramite a partire dai parametri calibrati?

7 MATLAB Codes

In this section we attach the *MATLAB* we used to obtain this results.

7.1 Inverse Gaussian Variable simulation

This code generates an arbitrare number of i.i.d. Inverse Gaussian variables with density function:

$$p(x) = \sqrt{\frac{\lambda}{2\pi x^3}} e^{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}} \mathbb{I}_{x\geq 0}$$

```
function X = igrnd(lambda,mu,n,m)
% Genera una matrice n x m di numeri casuali distribuiti secondo una
% INVERSE GAUSSIAN usando l'algoritmo proposto in Cont-Tankov
%
% lambda: primo parametro
% mu: secondo parametro

% Numero totale di simulazioni
nSim = n*m;

% Genera i numeri secondo una normale
N = randn(nSim,1);

% Y = N^2;
Y = N.^2;

% X
X = mu + mu^2.*Y./(2.*lambda) - mu./(2*lambda) ...
    .*sqrt(4*lambda*mu.*Y + mu.^2.*Y.^2);

% Genera i numeri distribuiti secondo una uniforme
U = rand(nSim,1);

% Restituisci i numeri con la if condition
idx = U ≤ mu./(X + mu);

X(¬idx) = mu^2./X(¬idx);

X = reshape(X,n,m);
end
```

7.2 Normal Inverse Gaussian process simulation

This function generater an Inverse Gaussian process.

```
function X = NIGSimulation(theta,sigma,k,T,nSim,nDates)
% Genera un moto browniano con drift subordinato ad un inverse ...
    Gaussian,
% ovvero genera un processo di tipo NIG
%
% MG 25maggio2019

% Calcola il dt
dt = T/nDates;
```

```

% Setta i parametri dell'inverse Gaussian
lambda = dt.^2/k;
mu = dt;

% Genera gli incrementi "temporali"
dS = igrnd(lambda,mu,nSim,nDates);

% Genera le normali (per il browniano)
N = randn(nSim,nDates);

% Genera gli incrementi
dX = sigma.*N.*sqrt(dS) + theta.*dS;


% Tira la traiettoria del processo
X = [zeros(nSim,1) cumsum(dX,2)];

end
```

7.3 Normal Inverse Gaussian Characteristic Function

This function valutate the characteristic function of a NIG process.

```
function v = phi_nig(u,S0,r,omega,T,theta,k,sigma)
% function v = phi_nig(u,S0,r,omega,T,theta,k,sigma)
% Normal Inverse Gaussian (NIG) Characteristic Funcion
% S0: starting value
% r: risk free
% omega: correcting parameter
% T: time to maturity
% k: subordinator parameter
% sigma: volatility

cfNIG = exp(T./k.*(1 - sqrt(1-2.*1i.*u.*theta.*k + ...
    u.^2.*sigma.^2.*k)));

v = exp((log(S0) + (r + omega) * T).*1i.*u).*cfNIG;


end
```

7.4 Fast Fourier Transform

This code perform a FFT pricing for an arbitrary characteristic function.

```
function [CallPrices,LogStrikes] = FFTPricing(T,r,phi)
% [CallPrices,Strikes] = FFTPricing(S0,K,T,phi)
% Pricing Call values using Method prodivided by Carr-Madan 1999
% T: maturity
% r: risk free
% phi: Characterist function of Log-prices
% For black-Scholes Framework that is
% phi=@(x) exp(1i*(log(S0) + (r-0.5*sigma^2)*T).*x ...
    -0.5*sigma^2*T.*x.^2);
%
% MG: 02marzo2017

% Integer (needed to define quadrature nodes)
L = 12;

% N, power of 2
N = 2^L;

% integration step
dv = 0.25;
% dk integration step on log(K) grid
dk = 2*pi/(N*dv);

% Alpha
alpha = 0.75;

% Vector v (integrands)
jvec=1:N;
v = (jvec-1)*dv;

% vector k (log strike)
uvec = 1:N;

% lower bound for ku
b=(N * dk) / 2;
ku=-b + dk*(uvec-1);

% To transform....
psi= exp(-r*T) .*phi(v-(alpha+1)*1i)./(alpha^2+alpha - v.^2 ...
    +1i*(2*alpha+1).*v);

% Simpson Quadrature Rule
SimpsonW=zeros(1,N) ;
SimpsonW(1)=1/3;
for i =2:N

    if(mod( i ,2)==0)
        SimpsonW( i )=4/3;
    else
        SimpsonW( i )=2/3;
    end
end
tmp=dv.*psi .* exp(1i * v *b ).*SimpsonW;


% Call Pricing
CallPrices=exp(-alpha.*ku)./pi .* real(fft(tmp));
LogStrikes=ku;


end
```

7.5 Margins Calibration Function

This function is passed to a minimization algorithm to obtain the parameter of the margins of a multivariate Norma Inverse Gaussian process $\boldsymbol{X}(t)$.

```
function fun = CalibFunctionNigTankov(x,S0,T,r,K,P)
% Calibra i parametri di un processo Normal Inverse Gaussian ...
    nell'ipotesi
```

```
% di essere sotto la misura Risk-Neutral

theta = x(1);
sigma = x(2);
k = x(3);

% Set drif condition for risk-neutrality
omega = -1/k.*(1 - sqrt(1 - 2*theta*k-sigma^2*k));

% Get distinct Maturities
Tdistinct = unique(T);
Nmaturities = length(Tdistinct);


for i = 1:Nmaturities
    % Get maturities
    idx = T == Tdistinct(i);


    % perform pricing via FFT
    % Call FFT Method
    [callPrices,logStrikes] = FFTPricing(Tdistinct(i),r,...
        @(w)phi_nig(w,S0,r,omega,Tdistinct(i),theta,k,sigma));

    % Get call Prices corrisponding to given prices
    K_sel = K(idx);

    % Prezzo delle Call
    call_Prices = interp1(logStrikes,callPrices,log(K_sel));

    % Funzione da minimizzare
    fun = call_Prices - P(idx);

end




end
```

7.6 Correlation Matrix Fitting

Minimizing function getCommonParametersLSQNONLINE.m we can fit the correlation matrix. The function is subject to some constraints computer by the function mycon.m

```
function f = ...
    getCommonParametersLSQNONLINE(x,corrNum,sigma1,theta1,k1,...
    sigma2,theta2,k2)
% Function to minimize to set correlation

% Get parameters from x
a1 = x(1);
a2 = x(2);
betaz = x(3);
gammaz = x(4);
nuz = x(5);

% Compute the paramtetic correlation
```

```
m1 = a1*a2*(gammaz^2 + betaz^2*nuz)/...
    sqrt((sigma1^2 + theta1^2*k1)*...
    (sigma2^2 + theta2^2*k2));

% Set difference
f = corrNum - m1;


end
```

```
function [c,ceq] = mycon(x,theta1,sigma1,k1,theta2,sigma2,k2)
% Non linear constraits to satisfy in order to obtain positive ...
    parameters
% for the NIG process.
%
%
% MG 29giugno2019

a1 = x(1);
a2 = x(2);
%betaz = x(3);
gammaz = x(4);
nuz = x(5);

c = [a1^2*gammaz^2 - sigma1^2;...
    a2^2*gammaz^2 - sigma2^2;...
    -nuz+k1; ...
    -nuz+k2];
ceq = [];

end
```

A The Variance Gamma Case

It can be shown that the approach adopter for the Normal Inverse Gaussian case can be easly definite for the Variance Gamma process to. For thi purpose it is enoght to remember that the characteristic function of a Variance Gama process is given by:

$$\phi_X(u,t) = \left(1 - iu\theta k + \frac{\sigma^2}{2}u^2 k\right)^{-\frac{t}{k}}$$

and the risk neutrality drift condition is obtained setting:

$$\omega = \frac{1}{k}\log\left(1 - \theta k - \frac{\sigma^2}{2}k\right)$$

Also the Monte Carlo Algorithm can be easy adapter to the Variance Gamma case. It is eought to simulate increments such as they follow a Gamma process. For this purpose an Algorithm is presented in [2]. Otherwise it is possibile to use the *MATLAB* function `gamrnd`. Remember that a Gamma density in *MATLAB* has the following parameterization:

$$f(x|a,b) = \frac{1}{b^a\Gamma(a)}x^{a-1}e^{-\frac{x}{b}}$$

All the numerical analysis performered fot the Normal Inverse Gaussian case can be repeted for the Variance Gamma case and leads to same conclusions. The makter fitting in good (Figure 8), correlation between commodities is higher that in the previos models (Figure 9), the spread is not so wide (Figure 11) as before and the Monte Carlo Algorithm is stable (Figure 12). In Table 5 and in Table 6 we display the calibrated parameters for the Variance Gamma case. In Figure 13 some paths in VG model are plotted.

Parameter	**Germany**	**France**
θ	0.307	1.696
σ	0.329	0.293
k	0.024	0.006

Table 5: Marginal parameters calibration for the Variance Gamma model

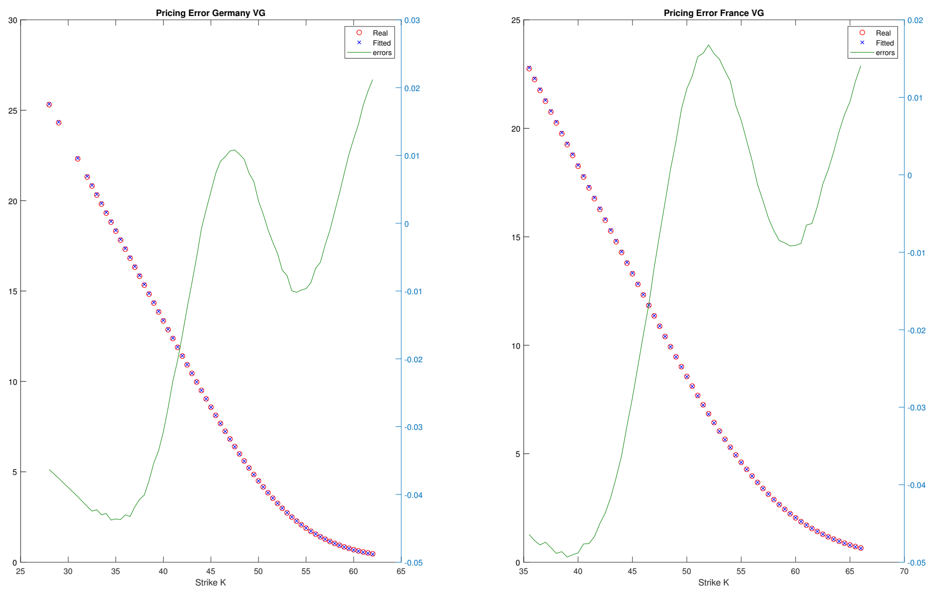


Figure 8: Market fitting of Call values on Forward for German and France market. We used the calibrated parameters to reprice Options quoted on the market. The re-pricing errori is very small. Variance Gamma model.

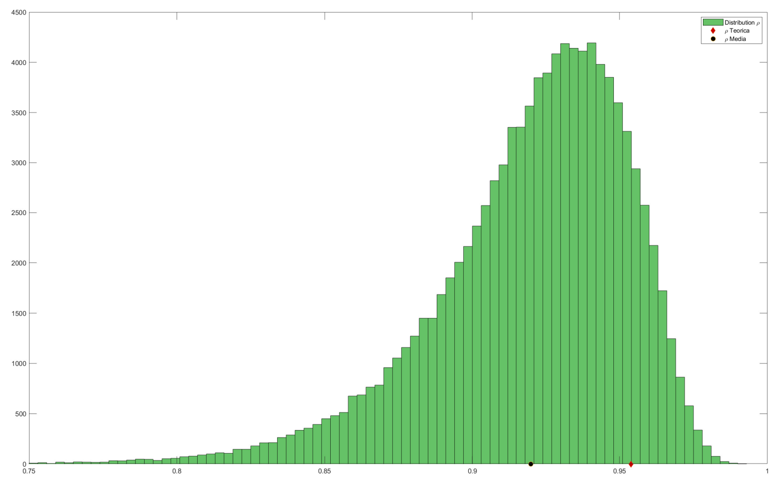


Figure 9: Log-returns correlation distribution. Variance Gamma model.

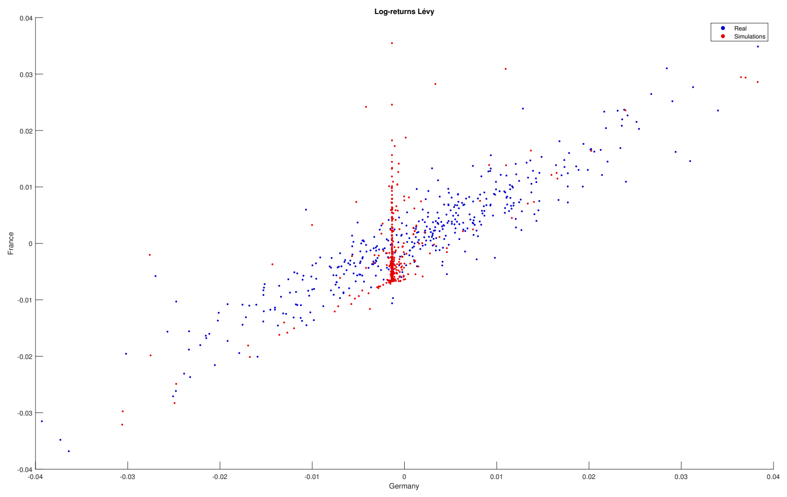


Figure 10: Log-returns scatter plot. In blue real data and in red simulation data. Variance Gamma model.

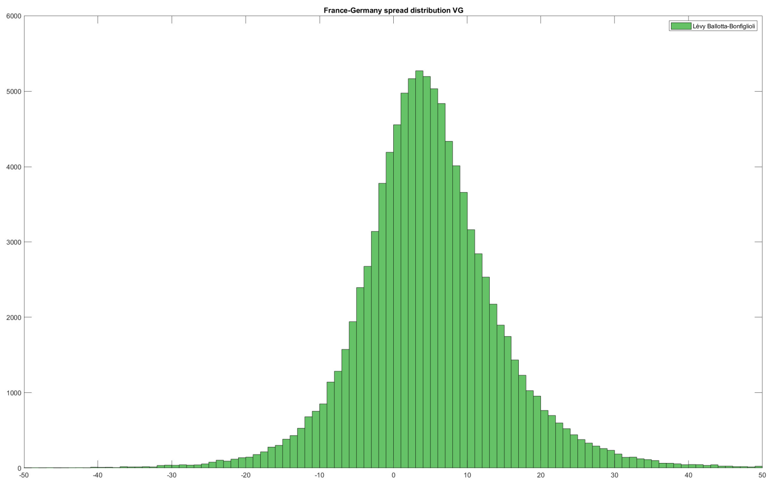


Figure 11: Spread Distribution. Variance Gamma model.

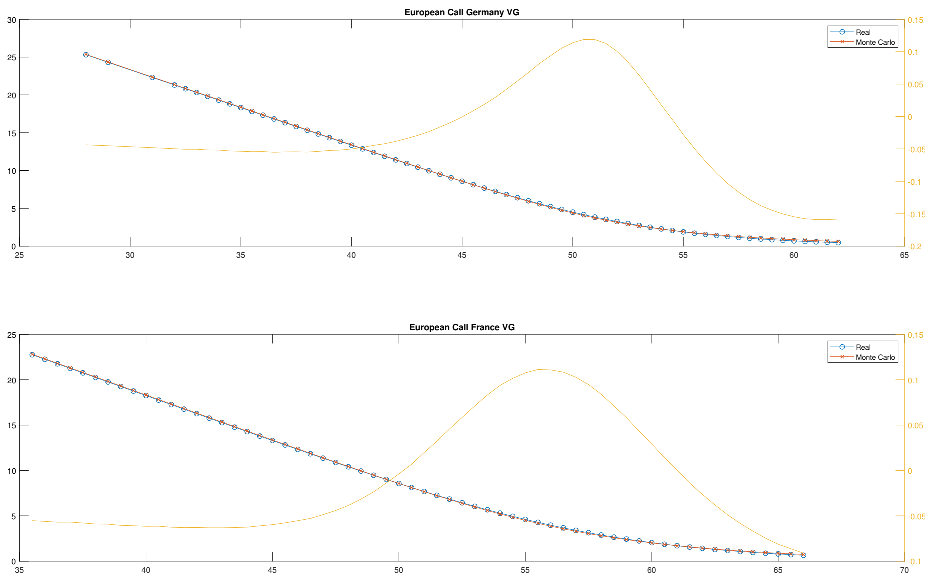


Figure 12: Repricing of the market using the Monte Carlo method. Variance Gamma model.

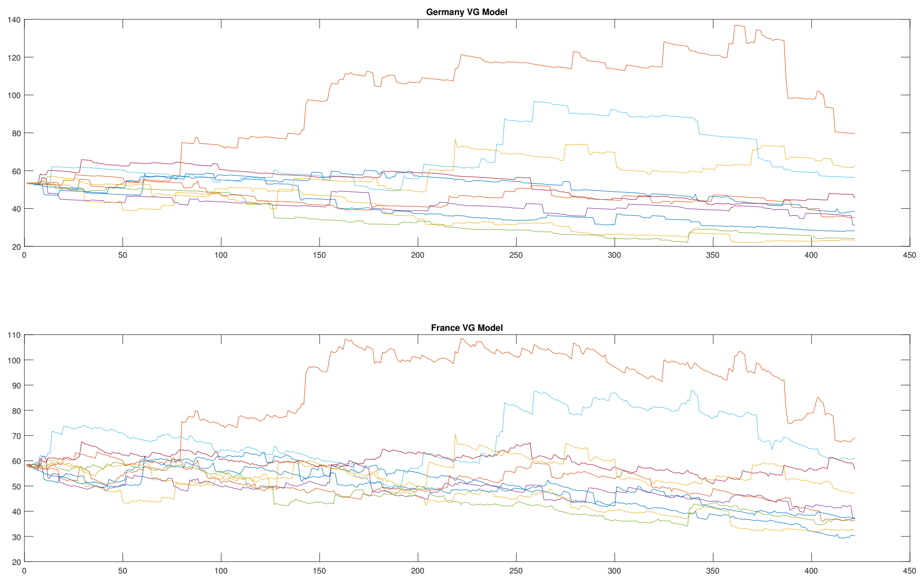


Figure 13: Some plots of simulated paths in Variance Gamma model.

Parameter	**Estimated Value**
a_1	0.147
a_2	0.131
β_Z	1.678
γ_Z	2.235
ν_Z	0.1
β_1	0.059
β_2	1.1476
γ_1	0.001
γ_2	0.001
ν_1	0.031
ν_2	0.006
$\rho_{Teorica}$	0.95
$\rho_{meanNumerical}$	0.92

Table 6: Marginal parameters calibration for the Variance Gamma model.

References

[1] L. Ballotta and E. Bonfiglioli. Multivariate asset models using Lévy processes and applications. 2013.

[2] P. Tankov and R. Cont. *Financial Modeling with Jump Processes*. Chapman and Hall, 2004.