Energy Options Calibration and Pricing (HJM NIG)

Energy Finance Final Project

Academic Year 2023-2024

Project 4

Davide Cantù
Tiziano Cinciripini
Giorgio Pietro Frigerio
Erik Gustav Malmsten
Luca Emanuele Targa

POLITECNICO MILANO

POLITECNICO
MILANO 1863

Contents

1. Introduction

In the intricate domain of Energy Finance, the meticulous calibration and pricing of exotic options associated with commodity and electricity swaps assume a central role in the realms of risk management and strategic financial decision-making. As the energy markets undergo continual transformation, the intricate nature of financial instruments linked to underlying assets, particularly commodity and electricity swaps, introduces both challenges and prospects for discerning market participants.

The aim of this project is to exploit the HJM NIG model on French electricity swaps in order to calibrate and price some structured payoff options.
The data we had to deal with were provided through an Excel file with 4 sheets. The first sheet contains all liquid maturities for French power swaps; the second and third sheets contain the option prices (implied volatility quotes) on the 2025 and the 2027 futures (monitoring period over the year, delivery at the end) on November 4th 2023; the fourth sheet contains the market discount factor curve.
The goals of this project are the following ones:

1. Understanding if the drift condition can be written explicitly
2. Finding an admissible range for the model parameters
3. Calibrating the model on the 2025 French option prices (the entire surface) by minimizing the distance between model prices and market prices
4. Pricing an option with payoff at time $t = 1$: $\left(\max_{t \in (0,1)} F(t, \tau_1, \tau_2) - K\right)^+$, where $F(t, \tau_1, \tau_2)$ identifies the swap price at time t with delivery period $[\tau_1, \tau_2]$, $K = 300$, τ_1 is January 1st 2025 and τ_2 is December 31st 2025.
5. Calibrating the model on both 2025 and 2027 option prices together
6. Pricing an option with payoff at time $t = 1$: $(\max \{F(1, \tau_1, \tau_2), F(1, \tau_3, \tau_4)\} - K)^+$, where τ_3 is January 1st 2027 and τ_4 is December 31st 2027

In order to achieve the stated goals, we largely exploited the energy finance book "Stochastic Modelling of Electricity and Related Markets" [1].
The report is organized as follows: the first two points are discussed in a unique section 2, in which a model overview is provided as well. In the following chapter 2025 options calibration is explored with all the details of the routines employed. The next section discusses the pricing of the path dependent option (point 4 above). Subsequently, we have another chapter about calibration (2025 and 2027 options), followed by a chapter on the pricing described at point 6 above. In all these chapters we are going to provide some limitations of the assumptions that we had to rely on, besides some possible improvements to get better results in terms of reliability.

2. Drift Condition & Model Parameters Constraint

In this section we start by introducing the model we are going to exploit throughout all the project and we focus on the first two aims. Namely, we try to understand if the drift condition, which guarantees the martingality of the model, can be written explicitly (or if we have to exploit some numerical techniques) and we find the constraints for the model parameters using the drift condition previously derived.

2.1. Model Presentation: HJM Model

We start our analysis by introducing the HJM model presented in [1].
The HJM model for forward contracts under the risk-neutral measure $\mathbb{Q}$ is described in the equation (6.1) of [1], which is the following one:

$$f(t,\tau) = f(0,\tau)\exp\left\{\int_0^t a(u,\tau)du + \sum_{k=1}^{p}\int_0^t \sigma_k(u,\tau)dW_k(u) + \sum_{j=1}^{n}\int_0^t \eta_j(u,\tau)dJ_j(u)\right\} \tag{1}$$

Where a, σ_k and η_j are real-valued (σ_k also positive), continuous functions on $[0,T]\times[0,T]$. T is an upper bound for the delivery times in the market.
As stated in [1], *The electricty and gas markets trade in forward contracts having a delivery period, for which we here will use the common notion swaps. The owner of a swap contract with delivery over the tyme* $[\tau_1,\tau_2]$ *would receive a constant flow of the commodity over this period, against a fixed payment per unit.*
Then, in our project we used the model for the swap contracts, which in case of HJM is the (6.9) of [1]:

$$\begin{aligned} F(t,\tau_1,\tau_2) = F(0,\tau_1,\tau_2)\exp\Bigg\{&\int_0^t A(u,\tau_1,\tau_2)du + \sum_{k=1}^{p}\int_0^t \Sigma_k(u,\tau_1,\tau_2)dW_k(u) \\ &+ \sum_{j=1}^{n}\int_0^t Y_j(u,\tau_1,\tau_2)dJ_j(u)\Bigg\} \end{aligned} \tag{2}$$

Where A, Σ_k and Y_j are real-valued (Σ_k also positive), continuous functions on $[0,T]\times[0,T]$. T is an upper bound for the delivery times in the market, and $0 \le t \le \tau_1 \le \tau_2 \le T$.
For what concerns our project, we employed a HJM model driven by a a NIG process, which is in facts the previous model with $p = 0$ and $n = 1$. Then, the model becomes the following one:

$$F(t,\tau_1,\tau_2) = F(0,\tau_1,\tau_2)\exp\left\{\int_0^t A(u,\tau_1,\tau_2)du + \int_0^t Y(u,\tau_1,\tau_2)dJ(u)\right\} \tag{3}$$

This means we have just one NIG driving process J. Moreover, we considered $Y(t,\tau_1,\tau_2)$ as a constant parameter Y, which we have to calibrate.
Then we have:

$$F(t,\tau_1,\tau_2) = F(0,\tau_1,\tau_2)\exp\left\{\int_0^t A(t,\tau_1,\tau_2)du + Y\int_0^t dJ(u)\right\} \tag{4}$$

But we know that $\int_0^t dJ(u) = J(t)$, then the final model is:

$$F(t, \tau_1, \tau_2) = F(0, \tau_1, \tau_2) \exp\left\{ \int_0^t A(t, \tau_1, \tau_2) du + Y J(t) \right\} \tag{5}$$

2.2. Drift Condition

For what concerns the drift condition, we can exploit the equation (6.10) of [1]:

$$\begin{aligned} \int_0^t A(u, \tau_1, \tau_2) + \frac{1}{2} \sum_{k=1}^{p} \Sigma_k^2(u, \tau_1, \tau_2) du + \sum_{j=1}^{n} \int_0^t Y_j(u, \tau_1, \tau_2) d\gamma_j(u) \\ + \int_0^t \int_{\mathbb{R}} e^{Y_j(u,\tau_1,\tau_2)z} - 1 - Y_j(u, \tau_1, \tau_2) z \mathbb{I}_{|z|<1} \nu_j(dz, du) = 0 \end{aligned} \tag{6}$$

which in case of our framework reduces to:

$$\int_0^t A(u, \tau_1, \tau_2) du + \int_0^t Y d\gamma(u) + \int_0^t \int_{\mathbb{R}} e^{Yz} - 1 - Y z \mathbb{I}_{|z|<1} \nu_j(dz, du) = 0 \tag{7}$$

At this point we can make use of the Lévy-Kintchine representation of the NIG Lévy process, as it is done in chapter 9 of [1]. Hence, we get the following expression:

$$\int_0^t Y d\gamma(u) + \int_0^t \int_{\mathbb{R}} e^{Yz} - 1 - Y z \mathbb{I}_{|z|<1} \nu_j(dz, du) = t \Psi_{NIG}(-iY) \tag{8}$$

Consequently, since we have an analytical expression of the Characteristic Function of the NIG process, the previous two equations lead to an analytical formulation for the drift condition:

$$\int_0^t A(u, \tau_1, \tau_2) du = -t \Psi_{NIG}(-iY) \tag{9}$$

For this reason, we do not need to compute any integral numerically.

2.3. Parameters Constraint

The cumulant function of the NIG process is the following one:

$$\Psi_{NIG}(u) = \frac{1}{\kappa} \left(1 - \sqrt{1 + u^2 \sigma^2 \kappa - 2iu\theta\kappa} \right) \tag{10}$$

Thus, making some computations, starting from 9, we get:

$$\begin{aligned} \int_0^t A(u, \tau_1, \tau_2) du &= -t \Psi_{NIG}(-iY) \\ &= -\frac{t}{\kappa} \left(1 - \sqrt{1 + (-iY)^2 \sigma^2 \kappa - 2i(-iY)\theta\kappa} \right) \\ &= -\frac{t}{\kappa} \left(1 - \sqrt{1 - Y^2 \sigma^2 \kappa - 2Y\theta\kappa} \right) \end{aligned}$$

From this computation we notice that we are able to directly calibrate the integral of A. Furthermore, the final condition on the model parameters can be found imposing the argument of the square root greater than or equal to 0, which leads to:

$$Y^2\sigma^2\kappa + 2Y\theta\kappa \leq 1 \tag{11}$$

Finally, we have other constraints concerning σ and κ. In particular, since σ represents an average volatility and κ its variance (sometimes called volatility of the volatility, or volvol), both of them have to be non-negative:

$$\sigma \geq 0 \tag{12}$$

$$\kappa \geq 0 \tag{13}$$

3. Calibration 2025 Options

In order to price derivatives, it is crucial to find a suitable set of parameters which properly describe the dynamics of the underlying asset. In our case, HJM NIG, the cumulant function which we employed in the pricing of plain vanillas in the Carr - Madan (FFT) framework is the following one (we now consider it already multiplied by t):

$$\Psi(u) = \frac{t}{\kappa}\left(1 - \sqrt{1 + u^2\sigma^2Y^2\kappa - 2iu\theta\kappa Y}\right) \tag{14}$$

thus, we had to calibrate a set of four parameters: $[\sigma, \theta, \kappa, Y]$.
Since we were given the implied volatility quotes of some call options, we first of all computed the market prices and stored them in a matrix. Subsequently, we created a matrix of function handle with model prices as functions of the model parameters and performed an optimization which minimized the distance between market and model prices. In order to do that, we used several methods to make our analysis as robust as possible.
In the following sections we present the results obtained with those methods and comment them.

3.1. Local Optimization and Least Squares

First of all, we tried to employ some classical MATLAB® tool. Namely, we exploited the fminsearch, fmincon and lsqnonlin functions to find the minimum distance, trying to use several distance definitions (L^1, to find results less sensitive to outlier, and L^2, to find results which minimize the RMSE).
The distance definition and the results of the three calibrations are summarized in the following paragraphs. Notice that, since we used the Carr - Madan FFT algorithm, we have some discretization parameters to be set. In order to get robust results, we calibrated for incremental values of those parameters (in the code they are A and Npow), using as starting point:

$$p_0 = [0.19\ 0.001\ 2.4\ 3]$$

3.1.1. fminsearch

In this case, the objective function to be minimized, is the L^1 distance between the market prices and the model prices. Namely, we used the following metrics:

$$d_1 = \sum_{i,j} |P_{mkt}(i,j) - P_{HJM}(i,j)| \tag{15}$$

where $P_{mkt}(i,j)$ is the market price corresponding to the i-th maturity and j-th strike and $P_{HJM}(i,j)$ is the corresponding model price.
Since fminsearch performs an unconstrained optimization, and we have the drift condition 9 and other constraints we explained above, we should check at the end if the constraints are fulfilled. Moreover, we used a trick to force the optimization to provide acceptable results; specifically, we added to the objective function the indicator functions multiplied by a very large number, where the sets of the indicator function are the sets where the constraints are not fulfilled.

Then, the objective function is:

$$f_{obj} = d_1 + M\left(\mathbb{I}_{Y^2\sigma^2\kappa+2Y\theta\kappa>1} + \mathbb{I}_{\sigma<0} + \mathbb{I}_{\kappa<0}\right) \tag{16}$$

where M is a very large number (e.g. 1e16).

The results provided by this kind of optimization are summarized in the following table with respect to some values of the discretization parameters A and Npow:

Discretization Parameters	Elapsed Time	RMSE	σ	θ	κ	Y
A = 400, Npow = 12	1.3126	14.6904	0.2173	0.0013	0.1451	2.6565
A = 600, Npow = 14	4.6164	14.6902	0.2173	0.0013	0.1451	2.6564
A = 800, Npow = 16	13.5038	14.6902	0.2173	0.0013	0.1451	2.6564
A = 1000, Npow = 18	69.1583	14.6902	0.2173	0.0013	0.1451	2.6565

Table 1: fminsearch calibration results

The results are stable as the discretization parameters grow. So, it seems that fminsearch is consistent with itself. Moreover, the RMSE seems to be acceptable for the complexity of our problem.
Before choosing to take these parameters as the result of the calibration, we want to check the results coming from other optimization algorithms.

3.1.2. lsqnonlin

In case of lsqnonlin, the optimization is a bit different. Indeed, for what concerns the code, it is sufficient to provide the matrix of differences between market prices and model prices, and the function minimizes the following objective function:

$$d_2 = \sum_{i,j} \left(P_{mkt}(i,j) - P_{HJM}(i,j)\right)^2 \tag{17}$$

We cannot include the non linear constraint related to the drift condition 9, but we can include the parameters bounds and check the condition at the end.
The results are summarized in the table below.

Discretization Parameters	Elapsed Time	RMSE	σ	θ	κ	Y
A = 400, Npow = 12	0.4895	13.6279	0.1949	0.0009	2.4000	2.9220
A = 600, Npow = 14	0.8445	13.6277	0.1949	0.0009	2.4000	2.9220
A = 800, Npow = 16	2.6689	13.6277	0.1949	0.0009	2.4000	2.9220
A = 1000, Npow = 18	15.4865	13.6277	0.1949	0.0009	2.4000	2.9220

Table 2: lsqnonlin calibration results

Even in this case, the results are very robust. It seems we took, as starting point, a point which is very close to a local minimum. However, these resulting parameters are similar to the ones computed with fminsearch with one exception: κ. Indeed, the values of σ, θ and Y are very close to the ones in table 1, but κ is more than ten times the previous one. It doesn't necessarily mean there were problems in the calibrations, since we have to consider two main aspects: first of all it is possible that our function presents several local minima. Secondly, we are comparing the results coming from the optimization of two different objective functions.
In order to make a check, we performed another minimization with fminsearch with the opti-

mization function 17. Unfortunately, we got different results and the elapsed time increased a lot. The reason of this behaviour is probably the different optimization algortihms carried out by fminsearch and lsqnonlin. Anyway, we avoid to report the results because are not so meaningful for our analysis.
The reason why there is a difference in κ is likely due to the different optimizations algorithms and to the presence of several local minima.
Finally, the RMSE is good and it is quite similar to the one of fminsearch. In particular, the ones with fminsearch are slightly higher, which is not surprising since the goal of lsqnonlin is exactly to minimize the MSE. So, we consider both results good enough.

3.1.3. fmincon

In the fmincon case, the optimization is the same as the fminsearch one, with the additional possibility of adding both the parameters' bounds and the non linear constraints. We used the same objective function and the results are the following ones:

Discretization Parameters	Elapsed Time	RMSE	σ	θ	κ	Y
A = 400, Npow = 12	14.5262	14.4798	0.0347	-0.1873	264.2968	168.5253
A = 600, Npow = 14	18.3802	14.5040	1.9402	-4.7021	47.8014	1.3177
A = 800, Npow = 16	161.3956	14.4818	0.0467	-0.2337	225.5163	115.8485
A = 1000, Npow = 18	493.2671	14.4863	4.4385	-16.2189	117.5495	0.8858

Table 3: fmincon calibration results

In this case the results are not satisfactory at all. First of all, they are not robust as the discretization parameters grow. Furthermore, in every single case is very difficult to interpret the resulting parameters. Finally, even if the RMSE is good, due to the above considerations and due to the huge elapsed time, we chose not to use these parameters for the pricing.

3.1.4. Discussion

In order to price financial derivatives having as underlying the swap price $F(t, \tau_1, \tau_2)$ we chose to take as reliable parameters the ones obtained employing the lsqnonlin routine. This choice is due to mainly two reasons: first of all they show a good stability in terms of discretization parameters. Moreover, they are the ones with lower root mean square error, so they are the ones which penalize the most large errors.
By taking them as model parameters, for a further investigation, we made a plot of the absolute differences between model prices and market prices. The obtained surface is in the image below (notice that we tried to make the same surface using the fminsearch parameters, however the plot was really similar. Thus, we avoid to report it). As it can be observed, despite the optimization with different routines, we got some peaks in which there is a big difference between model and market prices. This is probably due to several reasons. Firstly, it is possible that this model is not the best one in terms of reliability for this kind of asset. Secondly, the quality of the data does not seem particularly good. Hence, it is possible that this aspect affected the results. Indeed, if we look at the market prices computed simply by using the Black formula, one can easily observe that for each maturity, whenever the implied volatility passes the threshold of 1 (100%), the price increases even if the strike is higher. Specifically, for a given maturity we have options with higher strikes which worth more than other options with the same maturity but lower strikes. Since we are dealing with call options, this leads to

an evident arbitrage opportunity, hence, we cannot expect any good results for our analysis, as long as we work under the AoA[1] assumption.
The arbitrage technique allowed by the data is the following one. Let us denote two call options with C_1 (with strike K_1) and C_2 (with strike K_2) with $K_2 > K_1$ and $C_2 > C_1$ and same maturity T. I can buy C_1 and sell C_2 in $t = 0$, thus my profit in $t = 0$ is $C_2(0) - C_1(0) > 0$. At maturity T, the payoff of my position is:

$$(F(T, \tau_1, \tau_2) - K_1)^+ - (F(T, \tau_1, \tau_2) - K_2)^+ \geq 0$$

Since $K_2 > K_1$. So we are able to make a strictly positive profit in $t = 0$ with probability 1 and with probability 1 we don't loose anything in the future. This is the definition of arbitrage opportunity.

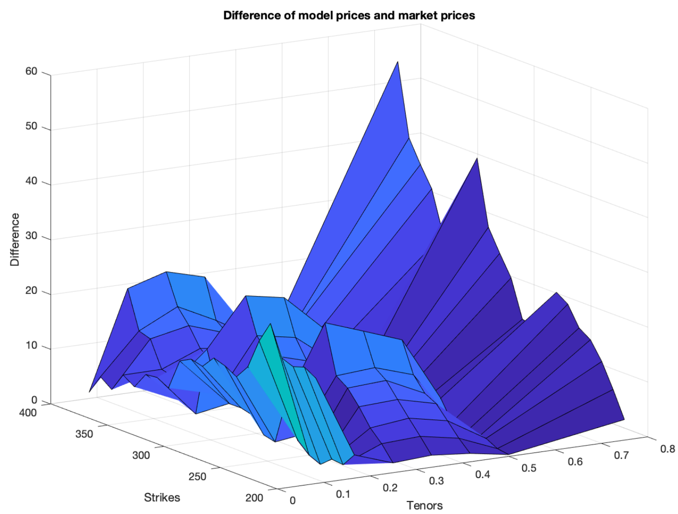

Figure 1: Absolute difference between model and market prices

3.2. Global Optimization: The Genetic Algorithm

Using the classical optimization tools, we encountered some problems. First of all, the parameters provided by the three different routines are a bit different, which suggests a poor robustness of our calibrations, since different optimization algorithms lead to very different solutions. Moreover, fmincon is not robust as the parameters A and Npow grow. Finally, the initial guess for the model parameters affects the optimal parameters found.
Consequently, this led to the fact that probably the objective functions we used could have a lot of local minima, and there is also a region where the function is almost "flat", since even with very different parameters, the errors are very similar.
In order to try to understand this problem, keeping in mind that the parameters provided by fminsearch and lsqnonlin are the most reliable ones, we searched for other methods which try to find the global minimum. For this reason we exploited the Genetic Algorithm, which is roughly introduced below, but first of all we would like to do a remark.
Remark. We are not using the Genetic Algorithm to find the optimal parameters. We are just using it to find a possible global minimum without any restriction in order to compute its RMSE and understand if the results provided by fminsearch and lsqnonlin are acceptable

[1]AoA: Absence of Arbitrage, i.e. we cannot generate profit from nothing, as it seems to be possible with the data we are dealing with.

or not. Since we are using as objective function the same as fminsearch, we could also find a higher RMSE with respect to the one of lsqnonlin. We just want to figure out the quality of the previous optmizations.

3.2.1. The Genetic Algorithm

Genetic algorithms are optimization algorithms inspired by natural selection processes. They are commonly used for solving optimization problems, aiming to find the best solution among a set of candidate ones.
The basic structure of the algorithm is the following:

1. Initial Population: Create an initial population of candidate solutions (individuals), where each individual represents a potential solution to the problem.
2. Fitness Function: Evaluate all individuals based on a fitness function that measures how well they satisfy the problem requirements.
3. Selection: Select individuals for reproduction based on their fitness scores; individuals with higher scores have a higher chance of being selected.
4. Crossover: Pairs of selected individuals undergo crossover to generate new individuals (offspring), simulating genetic recombination.
5. Mutation: With a certain probability, individuals may undergo mutation, resembling genetic mutations in nature.
6. Evaluation: Evaluate the fitness of the new individuals using the fitness function.
7. Replacement: Replace old individuals with new ones based on fitness scores.
8. Iteration: Steps 3-7 are repeated for a certain number of generations or until a stopping condition is met.

In order to use this algorithm on MATLAB® we had to install the add on "Global Optimization Toolbox", containing the function *ga*, which performs the algorithm.

3.2.2. Results

The results of the calibration carried out using the *ga* function are displayed in the following table:

Disscretization Parameters	Elapsed Time	RMSE	σ	θ	κ	Y
A = 600, Npow = 14	144.3141	14.4900	-12.0890	25.1281	34.9730	-0.1827
A = 800, Npow = 16	603.8170	14.5337	-12.0890	26.1405	34.4732	-0.1827

Table 4: fmincon calibration results

As expected, we notice that the calibrated parameters are totally meaningless, however this was not our aim. What it should be highlighted is that the RMSE corresponding to a point found with a global optimization algorithm is only slightly better than the ones obtained with fminsearch and a bit worse than the ones obtained with lsqnonlin (for the reason explained in the remark above).
To sum up all the considerations made in this chapter, we would say that the main problem

is that there are a lot of local minima, scattered across a vast region, which gives us many different set of parameters with very similar RMSE. This problem may arise since the data are not so good or they do not adapt well to our model. For example, for what concerns the data, we noticed that the provided volatility surfaces have many strange values, very far from what we are used to see. In particular, they vary between values near to 0 to values of 3 and above, which are very extreme.

4. Pricing Path Dependent Options

We now move on to the pricing of a path dependent (lookback) call option having as underlying the maximum value of a selected French electricity swap contract during a time horizon of 1 year, which is in facts the option's maturity. The payoff of such a contract is expressed as follows:

$$\left(\max_{t\in(0,1)} F(t,\tau_1,\tau_2) - K\right)^+ \tag{18}$$

where $F(t,\tau_1,\tau_2)$ is a swap having $\tau_1 = 1^{st}$ January 2025 and $\tau_2 = 31^{st}$ December 2025. The strike of our lookback option is K = 300. As stated in the previous section, we took the results from *lsqnonlin* algorithm as our model parameters in order to perform the pricing of the derivative according to a NIG driven HJM model. However, since we are dealing with a path dependent option, there are no available closed formulas for the pricing, meaning that we cannot rely on the usual Carr - Madan FFT algorithm. For this reason, we had to exploit Monte Carlo simulations in order to carry out this task. Since we had to run simulations, it would be better to firstly analyze the complete evolution of the employed model for the swap contract, by substituting the drift condition related to the characteristic exponent in formula (10) into the model equation stated in formula (5). Thus, we have:

$$F(t,\tau_1,\tau_2) = F(0,\tau_1,\tau_2)\exp\left\{-\frac{t}{\kappa}\left(1-\sqrt{1-Y^2\sigma^2\kappa-2Y\theta\kappa}\right)+Y\,J(t)\right\} \tag{19}$$

where $[\kappa,\sigma,\eta,Y]$ are indeed our calibrated parameters and J(t) is a NIG process.

Since J(t) is the stochastic process that drives the dynamics of the contract, we had to simulate the evolution of a NIG process. In particular, we performed the simulation of the ΔJ over a discretized time interval and then proceeded with the cumulative sum in order to obtain the values of J(t) depending on time. All the other terms in the model for the swap are instead deterministic, hence they do not require a stochastic simulation. After carrying out our Monte Carlo procedure and averaging the discounted payoffs thanks to $normifft$ function, we obtained the following results:

	Price	Left Extreme	Right Extreme
MC	36.3528	34.3409	38.3646
MC AV	35.8465	34.6421	37.0509

Noticing a large confidence interval for the regular Monte Carlo method (more than 10% of the price), we decided to perform the Antithetic Variables Monte Carlo procedure, in order to reduce its width (less than 7%). The amplitude of the confidence interval is likely due to an inadequacy of the considered model (probably too simplistic) for the dynamics of the swap contract. To verify this point, we performed several times this procedure with an increasing number of simulations and we obtained in all the cases very similar results. We can therefore conclude that the this pricing methodology has some limitations. Despite these problems, we checked if our outcomes were meaningful, thus we compared the price of this lookback option with the one of a plain vanilla call option written on the same contract. The result for the latter, derived through Carr - Madan FFT algorithm, is 27.6119. As expected, the plain vanilla price is significantly lower than the lookback one, since our path dependent option is based on the maximum value of the swap in a time horizon of 1 year, hence the probability of having a positive payoff in the lookback case is strongly higher with respect to the plain vanilla one.

5. Calibration 2025 and 2027 Options

The procedure for the calibration of the model using 2025 and 2027 options is the same as the one described in section 3. The only difference is that in this case, besides the difference between the model and market prices of the 2025 options, in the objective functions we have the difference between the model and market prices of the 2027 options as well.

5.1. Local Optimization and Least Squares

Even in this case, we employed the three different routines used for the calibration of the 2025 options. The followed approach is the same as in the previous chapter about calibration. Thus, we just report the results and then make a discussion at the end.

5.1.1. fminsearch

The results provided by the fminsearch function are in the following table:

Discretization Parameters	Elapsed Time	RMSE	σ	θ	κ	Y
A = 400, Npow = 12	3.5923	22.7437	0.3110	0.0013	0.0000	2.5506
A = 600, Npow = 14	9.4692	22.7433	0.3110	0.0013	0.0000	2.5505
A = 800, Npow = 16	34.1846	22.7433	0.3110	0.0013	0.0000	2.5505
A = 1000, Npow = 18	134.4678	22.7433	0.3110	0.0013	0.0000	2.5506

Table 5: fminsearch calibration results

Even in this case the parameters obtained through fminsearch are very stable as A and Npow grow. Moreover, the results are quite easy to interpret.

5.1.2. lsqnonlin

The results we got with lsqnonlin function are summarized in the table below:

Discretization Parameters	Elapsed Time	RMSE	σ	θ	κ	Y
A = 400, Npow = 12	23.5278	21.2510	0.5524	-1.2207	3.4314	2.1962
A = 600, Npow = 14	68.9458	21.2462	0.5334	-1.1883	3.7458	2.3378
A = 800, Npow = 16	208.8359	21.2458	0.7288	-1.6198	3.7735	1.7173
A = 1000, Npow = 18	670.4998	21.2581	0.3910	-0.8616	2.9907	2.9673

Table 6: lsqnonlin calibration results

For what concerns the robustness with respect to the discretization parameters, we have a stability which is weaker than the 2025 case and the fminsearch case. However, we can choose to take them as the results of the calibration anyway, but we have to pay attention on their reliability.

Remark. There are differences in the lsqnonlin parameters depending on the Matlab version. Namely, if one run the same code in a different machine, the values obtained may be slightly different.

5.1.3. fmincon

For what concerns thee fmincon routine, the results are the following:

Discretization Parameters	Elapsed Time	RMSE	σ	θ	κ	Y
A = 400, Npow = 12	9.6451	21.5027	0.7400	-8.7322	417.9905	11.7578
A = 600, Npow = 14	38.8289	21.5031	0.7289	-8.1953	378.3174	11.3549
A = 800, Npow = 16	137.7899	21.4964	1.0864	-17.9053	850.9366	11.4942
A = 1000, Npow = 18	811.0817	21.5008	2.6000	-99.000	4498.2000	11.0000

Table 7: fmincon calibration results

Once again, the results provided by fmincon are totally unacceptable. They are difficult to interpret and completely unstable with respect to A and Npow.

5.2. Discussion

The reliability of the calibration of the 2025 and 2027 options all together is arguable. Indeed, in principle there is no reason to assume that $F(t, \tau_1, \tau_2)$ and $F(t, \tau_3, \tau_4)$ follow the same dynamics, since they are two different assets. Hence, what we are asked to do is a simplification of the reality, and it is probably more appropriate to calibrate two different sets of model parameters. It is possible that the problems encountered in the calibration through lsqnonlin are caused by what follows from the issue discussed above. Indeed, when we calibrated only on 2025 options, we did not face this criticality.

6. Pricing Basket Options

We can now proceed with the pricing of a basket option, namely a derivative whose payoff depends on multiple underlying processes. In particular, in our case, we considered a call option on the maximum between two French electricity swap contracts. Thus, the resulting payoff is given by:

$$\left(\max_{t\in(0,1)}\{F(t,\tau_1,\tau_2),F(t,\tau_3,\tau_4)\}-K\right)^+ \tag{20}$$

where $F(t,\tau_1,\tau_2)$ is a swap contract with $\tau_1 = 1^{st}$ January 2025 and $\tau_2 = 31^{st}$ December 2025, $F(t,\tau_3,\tau_4)$ is another swap contract having $\tau_3 = 1^{st}$ January 2027 and $\tau_4 = 31^{st}$ December 2027, while the strike K is equal to 300.

6.1. Closed Formula under strong assumptions

Assuming the same model for the evolution of the two swap contracts, which are actually two different products and therefore have in reality a diverse stochastic evolution, we exploited the combined calibration on the options written on both 2025 and 2027 swaps. With this strong hypothesis, we came up with a single set of parameters, hence a unique model, to express the evolution of both contracts, which can be visualized below:

$$F(t,\tau_1,\tau_2) = F(0,\tau_1,\tau_2)\exp\left\{-\frac{t}{\kappa}\left(1-\sqrt{1-Y^2\sigma^2\kappa-2Y\theta\kappa}\right)+YJ_1(t)\right\} \tag{21}$$

and

$$F(t,\tau_3,\tau_4) = F(0,\tau_3,\tau_4)\exp\left\{-\frac{t}{\kappa}\left(1-\sqrt{1-Y^2\sigma^2\kappa-2Y\theta\kappa}\right)+YJ_2(t)\right\} \tag{22}$$

where the set of parameters $[\kappa,\sigma,\eta,Y]$ is the same for both equations. As in section 4, we relied on the results of the lsqnonlin optimization algorithm to derive the required parameters. At this point, in order to employ a closed formula for the pricing of our option, we need to base our analysis on even stricter assumption, namely that the stochastic parts of $F(t,\tau_1,\tau_2)$ and $F(t,\tau_3,\tau_4)$ (the exponentials) are two indistinguishable processes. Thus, we consider $J_1(t)$ and $J_2(t)$ in (21) and (22) as the same NIG process. With this hypothesis, we are able to find deterministically the maximum of the two swaps by simply observing the higher value between the two current prices ($F(0,\tau_1,\tau_2)$ and $F(0,\tau_3,\tau_4)$), since the exponentials assume at each time instant the same value in our simplified framework.

As a consequence, we can price our basket option simply as a call written on the maximum between $F(0,\tau_1,\tau_2)$ and $F(0,\tau_3,\tau_4)$, as the stochastic evolution is unique.

Thus, we exploited Carr - Madan FFT algorithm considering as initial value the biggest one between the current prices of the swap contracts in the derivative. Furthermore, we also decided to run Monte Carlo simulations as an alternative pricing strategy, having to simulate the evolution of only one NIG process, and then compare the two outcomes.

Results are as follows:

FFT Price	MC Price	Left Extreme	Right Extreme
36.8345	36.7782	36.5790	36.9773

We can notice how FFT Price and MC Price are close to each other, meaning that the model is coherent in its own, althought its reliability its very limited, due to the fact that it could seem over-simplified.

6.2. Monte Carlo Simulation

In order to deliver a more realistic model, we decided to price a basket option in another framework, where the processes $J_1(t)$ and $J_2(t)$ are no more indistinguishable, but are considered as two independent NIG processes. In this setting, we can not rely anymore on a closed formula according to Carr - Madan FFT algorithm, since we can not directly compute the characteristic function of the maximum between two NIG-driven processes. Thus, the only available pricing strategy is to perform Monte Carlo simulations. Once that we let $J_1(t)$ and $J_2(t)$ be two independent processes, we can also highlight that in a general framework they are characterized by two different sets of parameters, as they drive the stochasticity of two diverse contracts ($F(t,\tau_1,\tau_2)$ and $F(t,\tau_3,\tau_4)$). For this reason, we introduced a new kind of calibration, as explained hereafter. We had to derive the parameters needed in the dynamics of $F(t,\tau_1,\tau_2)$ and $F(t,\tau_3,\tau_4)$, which are now different in the two equations. Hence, we had to reformulate their dynamics as

$$F(t,\tau_1,\tau_2) = F(0,\tau_1,\tau_2)\exp\left\{-\frac{t}{\kappa_1}\left(1-\sqrt{1-Y_1^2\sigma_1^2\kappa_1-2Y_1\theta_1\kappa_1}\right)+Y_1J_1(t)\right\} \tag{23}$$

and

$$F(t,\tau_3,\tau_4) = F(0,\tau_3,\tau_4)\exp\left\{-\frac{t}{\kappa_2}\left(1-\sqrt{1-Y_2^2\sigma_2^2\kappa_2-2Y_2\theta_2\kappa_2}\right)+Y_2J_2(t)\right\} \tag{24}$$

where vector $[\kappa_1,\sigma_1,\eta_1,Y_1]$ comes from the calibration performed in section 3, namely the one only on options on 2025 swaps, while vector $[\kappa_2,\sigma_2,\eta_2,Y_2]$ is computed by following the exact same procedure as the one in section 3, but exclusively on options on 2027 swaps. With two different calibrations, we derived two sets of parameters needed in order to simulate the differently parametrized NIG processes $J_1(t)$ and $J_2(t)$. Once that we obtained the evolution of such underlyings, we could follow the usual Monte Carlo procedure by relying on $normfit$, leading us to the following results:

	Price	Left Extreme	Right Extreme
MC	40.5245	38.9696	42.0794
MC AV	40.2361	39.2207	41.2515

As it can be observed, the amplitude of the confidence interval for regular Monte Carlo is not extremely high (7.6% of the price), although it could be more satisfactory. For this reason, we once again performed the Antithetic Variables Monte Carlo procedure, reaching a thinner confidence interval (5%). Furthermore, if we compare these results with the price obtained by assuming that $J_1(t)$ and $J_2(t)$ are indistinguishable, we can notice that these prices are higher than the previous one. This is perfectly coherent with the theory, since by assuming that now $J_1(t)$ and $J_2(t)$ are two different NIG processes, we are introducing another source of randomness into our model, namely an additional quantity that could contribute to determine the maximum value between $F(t,\tau_1,\tau_2)$ and $F(t,\tau_3,\tau_4)$. This happens since with two different stochastic processes we can compute the actual $max\{\cdot\}$ between them at maturity, meaning that process $F(t,\tau_3,\tau_4)$ could be higher than process $F(t,\tau_1,\tau_2)$, while previously $F(t,\tau_3,\tau_4)$ was forced to be less than $F(t,\tau_1,\tau_2)$, since its initial value was the smallest one and all the stochasticity depended only on the evolution of a unique NIG process. Hence, the fact that the derivative has now more probability to be in the money at maturity justifies a higher price than before.

7. Conclusion

The goal of this project was to exploit the HJM NIG model on French electricity swaps in order to calibrate and price some structured payoff options.
At a first glance the results we obtained from our computations are not satisfactory at all. It seems we were not able to find global optimal parameters, in particular we got very different results by employing different calibration routines. Fortunately, at least the pricing seems to be coherent with what we were expecting, despite the quality of the calibration.
By further investigating the data we were dealing with, one can easily figure out that it was not trivial at all to get good results. As explained in the previous chapters, the implied volatility turns out to be very strange. By using the Black formula we got prices which do not fulfil the AoA assumption, leading to non reliable results, since all the theory we use in mathematical finance assumes that there are no arbitrage opportunities in the market.
Furthermore, the problems that we encountered could be justified by the fact that the chosen model is not so adequate for this task (probably too simplistic, p=0 and n=1 could be restrictive), while a more sophisticated one would be more appropriate. By taking into account all the above considerations, our results can be considered coherent with the theory. However, some improvements both on the quality of the volatility dataset and the choice of the model would lead to more reliable outcomes.

References

[1] Steen Koekebakker Fred Espen Benth, Jurate Saltyte Benth. Stochastic modelling of electricity and related markets. 2008.