Structuring and Hedging of a Structured Product

Partially Principal Protected Note

Vermeir Jellen

October 17, 2016

Contents

1 Introduction

In this paper we describe the structuring process of a Partially Principal Protected Note (PPPN). A PPPN is a structured product on an index and delivers one final payment to investors at maturity date T. The payoff structure at time T can be explained as follows:

$$\text{Payout} = \begin{cases} 95\%N + p * N * (\frac{S_T - S_0}{S_0})^+ & \text{if } B \text{ never hit} \\ 95\%N & \text{if } B \text{ is hit} \end{cases}$$

where N is the amount invested in the product, S_0 is the initial value of the underlying instrument at time 0, S_T is the value of the underlying at maturity and B is the barrier level of the product. The premium $p * N * (\frac{S_T - S_0}{S_0})$ is calculated as the initial investment N multiplied by the participation rate p and the positive index return over the period $[0, T]$. This premium is only paid when the stock price never went below B during the lifetime of the product.

The paper is structured as follows: We start out by giving a general high level overview of the structured product and its properties. The goal here is to sell the product to investors while providing all necessary information about its risk/return characteristics. Next, we offer a more technical explanation about the actual structuring process. We show that the product can be structured by splitting it in a bankaccount position and an exotic option position. Hence, In order to price our PPPN correctly we must obtain a fair price for the underlying exotic option. In this paper we perform the pricing under Heston and we calibrate the Model on the available index vanilla option market prices. A number of performance optimization and robustness techniques are also discussed in this context. Once we obtain satisfactory model parameters, we apply a monte carlo approach in order to obtain a fair price for the exotic. Additionally, we also analyze some relevant variance reduction techniques. In the third and final section of the paper we focus our attention on the hedging aspect of the product. We discuss a dynamic hedging approach and illustrate how to obtain a Greek neutral portfolio at inception.

2 Investment Analysis Report

2.1 Product Overview and Market Data

The product that we sell to the investors is a partially principal protected note with index XLB as the underlying. The derivative will be issued on May 19th 2015 and it will be sold for $N = 1000000\$$. The maturity date T of the product is set to be 1st of March 2017. The current stock price of XLB at time T_0 is 51.04 and the index offers an annualized dividend payout of 1.45%. The payout structure is defined as above, with the barrier set at 70% of the initial price at 35.7280\$. The participation rate p is defined to be 55.35%. The risk free bank account rate for maturity T is 0.51%. This rate and other rates can be obtained via cubic interpolation of the US yield curve. More information on the marketdata assumptions are provided in the appendix. For a full analysis on the calculation of the participation rate, the reader is referred to the subsequent structuring chapters.

2.2 Payoff Structure

From an investor point of view, there are a number of possible outcomes that can manifest itself at maturity time T. These outcomes will depend on both the barrier B and the final value S_T of the XLB index. Following scenario's and their respective payoff results can occur:

1. The barrier was never breached and $S_T > S_0$: Investor receives back 95% of the original investment plus a premium of $p * N * (\frac{S_T - S_0}{S_0})$ at time T.
2. The barrier was never breached and $S_T \leq S_0$: Investor receives back 95% of the original investment N at time T.
3. The barrier was breached during the lifetime of the product: Investor receives back 95% of the original investment N at time T.

In summary, the product offers a large protection to the downside risk of the index while participating in its upside potential ($p = 55.35\%$) as long as the barrier is not breached. The capital of the investor is always 95% protected.

2.3 Risk Return Profile

In this section we describe the properties of the investment return at maturity T in comparison with direct investment in the XLB index. The return at time T can be described as the total payoff of the product minus the initial invested amount N. Figure 1. Below illustrates the comparison. When invested in the PPPN, the investor obtains a certain loss of 50000USD when the barrier is breached during the lifetime of the product. In the scenario where the barrier is not breached, a premium will be added to the amount under protection. We again notice that the premium is only positive when the index price at maturity T is greater than the initial index price ($St > S0$). The points of interest on the graph are marked with circles: The purple circle indicates the price S_T below which the return of the derivative becomes greater than the return of the underlying index. Below this price the capital of the investor is protected while direct investment in the index could result in large potential losses. The red circle indicates the breakeven point for investment in the PPPN. If the final stock price at maturity is above this price then we will obtain an uncapped positive return that is linked to the final price S_T of the stock. The return on the upside is less than direct investment in the stock. This is the price the investor pays for receiving capital protection on the downside.

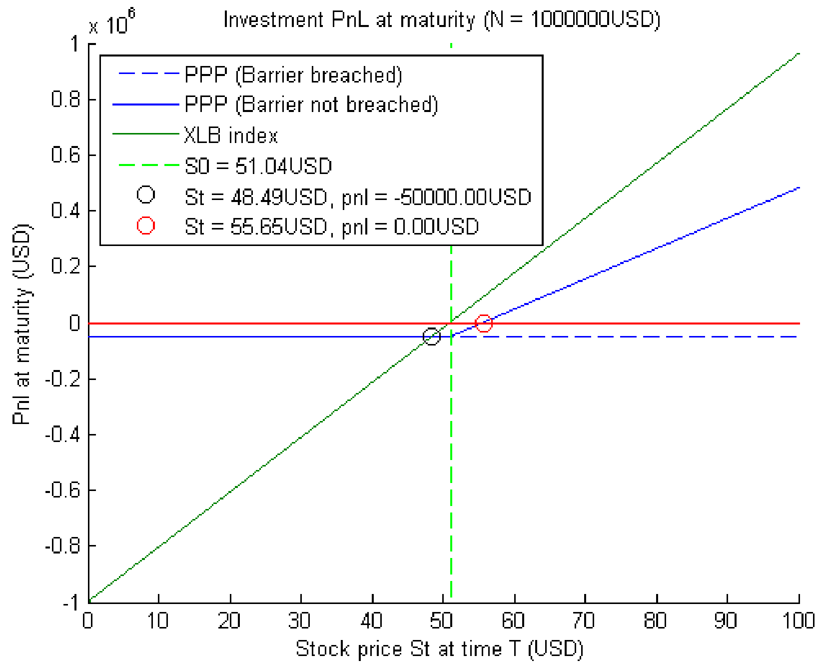

Figure 1: Investor PnL at maturity T (PPN versus direct Index Investment)

3 Technical Report

In this section we illustrate how the PPPN can be structured in such a way that the required payoff will be achieved at time T in all possible states of the world. We show that the structuring problem reduces itself to the pricing of an exotic barrier option and our goal will be to price this option under the Heston model. In this paper we focus on model calibration via Carr-Madan formulation in a well optimized and robust manner. Distinct choices for minimization criteria and weights are considered, early stopping and cross validation techniques are investigated. In the final subsection of this chapter we use the calibrated model to price the exotic option via monte carlo simulations. In this setting, 2 well known variance reduction techniques are analyzed: Stock path generation with antithetic random variables and variance reduction through an added control variate.

3.1 Structuring of the PPPN

In order to obtain the desired payout of the PPPN in all possible scenario's at time T we split the product in a bankaccount position and a position in down and out barrier calls with strike price S_0, barrier $B = 0.70 * S_0 = 35.73\$$, and maturity T. We also implement a profit margin of 1% of the invested amount

for services offered to the clients.
To distribute the money correctly, we first add a certain amount to the bankaccount in such a way that we can pay the guaranteed capital of $95\% * N$ to the client at maturity: $B(0) = 95\% * N * exp(-rT)$ with T the time until maturity expressed in years and r the corresponding risk free interest rate. We can then subtract our $Profit = N * 1\%$ from what is left and the remainder can be used to buy the DOBC options: $totalValueDOBC = N - B(0) - Profit$. More explicitly, expressed in numbers, we get:

- $B(0) = 95\% * 1000000\$ * exp(-0.51\% * 1.7863) = 941420\$$
- $Profit = 1000000 * 1\% = 10000\$$
- $totalValueDOBC = 1000000 - B(0) - 10000 = 48584\$$

We have already explained that the payoff of the bankaccount covers the capital guarantee. To understand how we cover the premium we note that the payoff of one DOBC option equals $(\frac{S_T - S_0}{S_0})^+$ when the barrier B is never hit during the index' lifetime, and 0 otherwise. One can easily understand that this payout corresponds to the premium payout of the PPPN. However, we still need to figure out how many DOBC options we can buy with the $totvalValueDOBC$ in order to obtain the participation rate. Unfortunately for us, DOBC options are generally not freely available on the market so we have to obtain a fair value price for the instrument ourselves. In the next subsections we illustrate how this can be achieved under the Heston model.

3.2 Pricing Vanilla Options under the Heston Model

3.2.1 Heston Model Specification

Under the heston model, the stock price and the (squared) volatility process in the risk neutral world can be described as follows:

$$\begin{cases} \frac{dS_t}{S_t} = (r-q)dt + \sqrt{v_t}dW_t, S_0 \geq 0 \\ dv_t = \kappa(\eta - v_t)dt + \theta\sqrt{v_t}d\tilde{W}_t, v_0 = \sigma_0^2 \geq 0 \\ dW_t d\tilde{W}_t = \rho dt \\ \kappa, \eta, \theta, v_0 > 0 \\ -1 < \rho < 1 \end{cases}$$

where W_t, $\tilde{W}_t$ are correlated Brownian motion processes for all $t \geq 0$, with correlation parameter ρ. v_t can be considered a square root mean reverting process with long run mean η and rate of mean reversion κ. θ represents the volatility of the variance, v_0 is the initial variance. Additionally the Feller condition should also be added to restrict the model from reaching negative volatility values: $2\kappa\eta \geq \theta^2$.

We now want to calibrate the Heston model on the available option prices and subsequently price our exotic option through monte carlo simulations. However, closed form solutions for vanilla options under Heston are not readily available. Luckily, Carr and Madan provide a way to price vanilla options under any model through utilization of the models' characteristic function. This Carr-Madan formulation will be explained in the next subsection.

3.2.2 Carr Madan Formulation

We will not provide a full derivation of the carr-madan formula and its implications here, but it can be shown that under certain conditions $v_j = \eta(j-1)$, $k_n = -b + \lambda(n-1), n = 1...N, where \lambda = \frac{2b}{N}$ the carr madan formulation can be written as follows:

$$\begin{cases} C(k_n, T) \approx exp(-\alpha k_n)\frac{1}{\pi}\beta_n \\ \beta_n = \sum_{j=1}^{N} exp(-\frac{i2\pi(j-1)(n-1)}{N})\alpha_j \\ \alpha_j = exp(iv_j b)\varrho(v_j)\eta_j s_j \\ s_j = (\frac{3+(-1)^j - \delta_{j-1}}{3}) \\ \varrho(v_j) = \psi(v_j; T) = \frac{exp(-rT)*\phi(v_j-(\alpha+1)i, T)}{\alpha^2+\alpha-v^2+i(2\alpha+1)v} \end{cases}$$

Here, ϕ represents the characteristic function of the model under consideration. The k_n correspond to the log strike prices of the options that are priced. The extra term s_j in the formulation represents the simpsons weighting rule that offers a more refined approximation of the integral that is approximated by the discrete summation. $\delta_{j-1} = 1$ if $j = 1$ and 0 otherwise. Carr and Madan report that values $\eta = 0.25$, $N = 4096$, $\alpha = 1.5$ give satisfactory results. This corresponds to $\lambda = 0.0061$ and $b = 12.57$. It is important to note that $(\beta_n, n = 1, ..., N)$ represents the fast fourier transform of vector $(\alpha_n, n = 1, ..., N)$. This computationally efficient transform must be processed only one time in order to compute prices for all the options with the same maturity T. Prices of options with strikes that do not exactly match the FFT prices can then be obtained via cubic interpolation. Application of Put-Call parity delivers us the corresponding put prices.

3.2.3 Heston Characteristic Function

If we can manage to plug in our Heston characteristic function in the Carr-Madan formula then we are ready to price vanilla options. The characterstic function of the Heston model is defined as follows:

$$
\begin{cases}
d = \sqrt{(\rho\theta ui - \kappa)^2 - \theta^2(-iu - u^2)} \\
g = \frac{\kappa - \rho\theta ui - d}{\kappa - \rho\theta ui + d} \\
A = iu(logS_0 + (r - q)t) \\
B = \eta\kappa\theta^{-2}((\kappa - \rho\theta ui - d)t - 2log(\frac{1 - g*exp(-dt)}{1-g})) \\
C = \frac{\sigma_0^2\theta^{-2}(\kappa - \rho\theta iu - d)(1 - exp(-dt))}{1 - g*exp(-dt)} \\
\phi_{u,t} = exp(A) * exp(B) * exp(C)
\end{cases}
$$

3.2.4 Computational Considerations

Two separate versions of the Carr-Madan formulation and the Heston characteristic function are implemented in Matlab. The entrance point of the first implementation can be found in "HestonPricerWithoutPrecomputation.m". This pricer takes a number of options as input and calls the "CarrMadanHeston" function one time for each separate maturity T. All options with the same maturity are priced simultaneously through a vectorized approach. The "CarrMadanHeston.m" and "HestonCharacteristic.m" contain the basic implementations of the Carr-Madan formulation and Heston Characteristic functions respectively, as described above.

"HestonPricer.m" is the entrance point towards the more performance enhanced implementation: It calls the underlying "CarrMadanHestonOptimzed.m" and "HestonCharacteristicOptimized.m" functions. To understand where the the performance gain is comming from one can take a look at "PrecomputationCarrMadanParameters.m" and" PrecomputationHestonCharacteristicFunction.m". The first file contains a number of Carr-Madan related variables that are precomputed one time, independent of the model. The second file contains a number of characterstic function related variables that can be precomputed for the Heston Model specifically: These parameters are independent of the options' maturity. As a result of this optimization process, only 2 exponential function calls and 1 logarithm call is required during the characteristic function evaluation. The number of necessary vectorized operations during the computation of the Carr-Madan formula is also significantly reduced.

3.3 Model Calibration

We now have the tools at our disposal to price vanilla options under the Heston model and our next aim is to calibrate this model to the available option prices in the market. However, this might not be an easy task given the complexity of the Heston model. The model contains 5 free parameters that must be calibrated simultaneously: $\kappa, \eta, \theta, \rho$ and σ_0. In order to fit such a complex model an efficient optimization algorithm must be applied to the problem at hand. Such an optimization algorithm also requires a minimization criterion to evaluate the datafitting process iteratively. Additionally, overfitting avoidance and robustness of the calibration should be kept in mind. In the next subsections we illustrate how we tackle these issues.

3.3.1 Constrained Optimization

In order to solve our problem it should be noted that we are able to make an educated guess for the initial Heston parameters. We can place some restrictions (upper/lower bounds) on them as welll. To obtain sensible results we assume the following set of constraints:

$$\begin{cases} 0 \leq 2\kappa\eta - \theta^2 \leq 20 \\ 0 \leq \eta \leq 1 \\ 0 \leq \theta \leq 5 \\ -1 \leq \rho \leq 0 \\ 0 \leq sigma_0 \leq 1 \end{cases}$$

We assume that a local minimizer will suffice to perform the optimization because our problem is bounded and fairly well defined. We choose here to tackle the problem with a constrained Nelder-Mead downhill simplex method. This method can be utilized to optimze non-linear optimization problems for which the derivatives are unknown. If the solution should fail to converge towards an acceptable solution then we can apply global search techniques such as direct search or simulated annealing. As we will see shortly, this is unnecessary.

In order to apply the Nelder-Mead optimization algorithm we must specify a well defined minimization function. In this paper we mainly focus our attention on the minimization of the weighted sum of squared errors between the calibrated prices and the option mid market prices:

$$criterion = \sum_{i=1}^{n} weight_i * ((calibratedPrice_{i-}midMarketPrice_i)^2)$$

where n is the number of avaliable market options. This functionality is implented inside the "MinimizationCriterionHeston.m" function: Given a set of Heston Parameters as input, the HestonPricer is called on a set of options and the resulting value of the criterion is returned as the final result. Note that the minimization of this criterion is equivalent to the minimization of the weighted RMSE. The true value of the RMSE between two sets of prices can be computed via "RMSE.m" function. Additionally, a similar function which is called "RMSESpreadAdjusted.m" was also implemented. This criterion assumes that prices that fall inside the bid ask spread contain no errors.
The only question that remains now is the choice of the weights. We consider three sensible options to weigh a particular options' error in the criterion:

1. Equal weighing: $W_i = 1$
2. Bid Ask weighing: $W_i = \frac{1}{abs(ask_i - bid_i)}$
3. Implied volatility weighing: $W_i = impliedVol(optionprice_i)$

The entrance point to the calibration process can be found in "HestonCalibrationLocalOptimization.m": In this function, the fminsearchbnd algorithm performs a constrained optimization for our minimizationcriterion in an iterative fashion. The constraints are defined as explained above and the desired weights should be given as input to the algorithm. There are some other optional inputParameters that will be explained in the next section.

Additionally, The function "BlackScholesImpliedVolatility.m" is available to compute the implied volatilities. This function determines both the the individual implied volatilities and the global implied volatility for a set of inputoptions.

3.3.2 Robustness and Overfitting Avoidance

When calibrating 5 free parameters it is easy to overfit a model on the in sample data while possibly obtaining bad out of sample results. To investigate this model overfitting risk we proceed as follows: We first split our optiondata in a training set and a testing set. Calibration is performed on the training data while the actual pricing validation occurs on the testset. In this paper we analyze 3 distinct types of calibration: Full convergence minimization, early stopping and cross validation. The three calibration procedures can be explained as follows:

1. Stop when full minimization or 10000 iterations reached.
2. Stop when $\sum_{i=1}^{n} w_i * [Cc_i(K_i, Ti) - Cm_i(K_i, Ti)]^2 \leq \sum_{i=1}^{n} w_i * [ask_i - bid_i]^2$
3. Stop when out of sample Adjusted RMSE remains stable or increases.

where Cc_i and Cm_i represent the calibrated prices and real market prices respectively. "StopFunction.m" is provided to fasciliate the early stopping technique, where the "globalStoppingCriterionBidAsk" is defined externally. During the cross validation technique we check the out of sample adjusted RMSE every 75 iterations and terminate the optimization process when it is stable or goes up again. The "HestonCalibrationLocalOptimization.m" allows us to feed early stopping and iteration settings as input.

Next, in order to compare the performance of the methods and the weight structures, we obtain prices for the out of sample options in three distinct ways:

1. Carr Madan Pricing
2. Monte Carlo Pricing Vanilla
3. Monte Carlo Pricing Exotic: $CALL = DOBC + DIBC$

The monte carlo exotic pricing entails the generation of random barrierlevels to generate DOBC and DIBC prices. These prices can then be added to obtain the call price of the out of sample option (or the put price via put call parity).

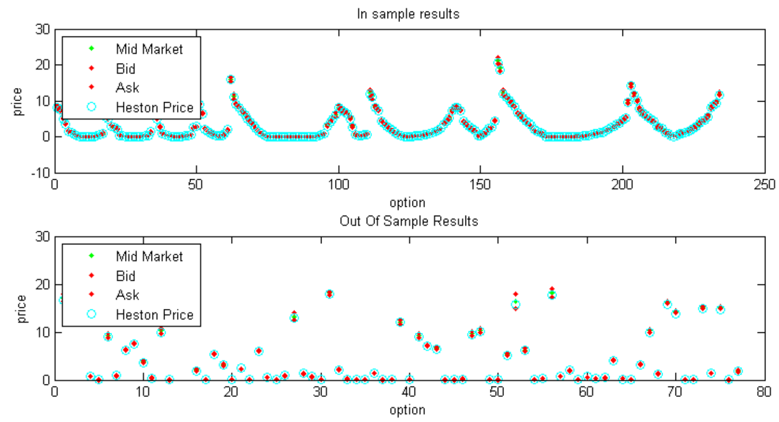

Figure 2: Calibrated Heston Model: In sample versus out of sample prices

Finally, we obtain the RMSE and adjusted RMSE values for the out of sample prices and compare results.The analysis is performed via "TestEarlyStoppingCrossValidation.m". Also view "MonteCarloSimulationRobustnessTest.m" for bulk option pricing using Monte Carlo. Results are reported in the appendix.

In the remainder of the paper we assume calibration via cross validation and Black-Scholes implied volatility weights. "PPPN.m" is the main entrance point of the PPPN pricer. In this script the "CalibrateHestonParametersCrossValidation.m" is called. In this function minimization of the out of sample adjusted RMSE is performed as discussed above and the optimal Heston parameters are returned. Because of the randomized nature of the cross validation, results can differ sligthly each run. An example calibrated parameterset is $\kappa = 3.0825$, $\eta = 0.0462$, $\theta = 0.5339$, $\rho = -0.8595$, $\sigma_0 = 0.1202$. A visualisation of the calibration result is shown in Figure 2.

3.4 Monte Carlo Exotic Pricing

Our Heston Model is calibrated and we are ready to price our exotic option via monte carlo simulations. Heston variance and stock paths can be simulated through the Milstein Scheme, as follows:

$$\begin{cases} S_{t+1} = S_t(1 + (r-q) * \Delta t + \sqrt{v_{t_i}}\sqrt{\Delta t}\epsilon_1) \\ v_{t_i+1} = v_{t_i} + (\kappa(\eta - v_{t_i}) - \frac{\lambda^2}{4})\Delta t + \lambda\sqrt{v_{t_i}}\sqrt{\Delta t}\epsilon_2 + \frac{\lambda^2 \Delta t(\epsilon_2)^2}{4} \end{cases}$$

where $\epsilon_1 = \epsilon$ and $\epsilon_2 = \rho\epsilon + \sqrt{1-\rho^2}\epsilon^*$ with ϵ and ϵ^* standard normal random numbers. This process is implemented in "SimulateStockPathsHeston.m". Note that we keep the variance process positive by taking absolute values if necessary. An alternative implementation is added in "SimulateStockPathsAntiThetic.m", where additional antithetic stock paths are generated.

The actual monte carlo simulation for an exotic DOBC is implemented in "MonteCarloPricerDOBCHeston". This function returns the price and upper and lower confidence intervals for a specific DOBC, given its properties as inputs. Depending on the variance reduction settings, following modes of operation are provided:

- Standard Monte Carlo simulations.
- Simulate antithetic pairs of the stock path and compute the resulting DOBC values in pairs. Calculate $E[Y] = E[\frac{X_1+X_2}{2}]$, where X_1 and X_2 represent the prices of the first and second elements of the generated pairs.
- Control variate technique: $E[X] = E[Y] = E[X - b(C - E[C])]$
- A combination of the above two techniques.

Variance reduction via the antithetic pairs technique can be intuitvely explained by the fact that the sample variance $s^2(m) = \frac{1}{m-1}\sum_{j=1}^{m}(Y_j - E[Y_m])^2$ is reduced because each Y_j is constructed out of two negatively correlated elements. The overal standard devation is also lower because we have more generated paths ($n = 2m$). We get $E[Y(m)] \pm z_{\frac{\alpha}{2}}\frac{s(m)}{\sqrt{n}}$ as the $100(1-\alpha)\%$ confidence interval. In contrast, the control variate technique is based on the observation that $\sigma_Y^2 = Var(Y) = \sigma_X^2 + b^2\sigma_C^2 - 2b\sigma_{X,C}$. Hence, by choosing C and b wisely, it is possible to reduce the overall variance. To keep it short, it can be shown that the optimal $b = \frac{\sigma_{X,C}}{\sigma_C^2}$; If we substitute this value in the previous equation then it becomes obvious that the reduction in total variance corresponds to the correlation between X and control variate C. In this paper we have chosen the terminal value of the stock price at maturity ($C = S(T)$) as the control variate, mainly because its value is part of the simulation already. The runtime performance and variance reduction of the techniques have been tested via "TestMonteCarloVarianceReductionAndStockSimulations.m". Results are reported in the appendix.

In the PPPN structuring context we use a combination of antithetic and control variate settings to price the DOBC options. The final price lies around 4.50USD per option, with a very low 95% confidence interval of 0.02USD. It should be noted that depending on the "PPPN.m" run, the average price might fluctuate a cent or 2. However, this is not related to stability problems of the Monte Carlo runs, it is related to the fact that we obtain slightly different Heston parameters through the randomized cross validation process. This is well explained by the fact that we use the adjusted RMSE as the stopping criterion and bid ask spreads are very wide for the later maturities. If we repeat the monte carlo analysis inside the same run of the PPPN.m then we notice that prices are very stable and fall inside the confidence intervals with very low fluctuations.

3.5 Structuring of the PPPN (Revisited)

Now that we have determined the price of our underlying down and out barrier call, we are ready to finish the structuring process. At time T_0 the amount of options that can be bought is $\frac{totalValueDOBC}{DOBCPrice} = \frac{48584}{4.5} = 10796$. To calculate the participation rate we compare the amount of stocks that could have been bought with the original notional versus the amount of DOBC in our PPPN:

- $nrStocksNotional = \frac{notional}{S_0} = \frac{1000000}{51.04} = 19592$

- $participationRate = \frac{nrDOBC}{nrStocksNotional} = \frac{10796}{19529} = 0.5510$

We now have all the information that we need. This concludes the structuring of our PPPN.

4 Hedging of the Greeks

In order to dynamically hedge the PPPN we must create offsetting positions in such a way that the total Greek sensitivities of the hedged portfolio becomes zero. In this section we demonstrate the hedging process at time T_0 in order to create such greek neutral portfolio's. Future rebalancing might still be necessary. We assume buying and selling of the offseting positions at the current mid market prices. In practice, bid and ask prices should be used to execute the transactions.

4.1 Greek Exposures

The sold PPPN is split up in a bankaccount position and a DOBC position. We do not hedge our bankaccount position because the final payoff is deterministic and without risk (We do not take ρ sensitivity into account). Hence, we focus our attention on the hedging of the DOBC position.

Hedging positions are obtained in the bottom part of "PPPN.m". We first calibrate our underlying DOBC under the Black-Scholes model via "BlackScholesImpliedVolatilityExotic.m", given the Heston model price as input. Next we obtain the corresponding Black-Scholes Greeks of our portfolio. We obtain the Greeks for our selected vanilla hedging instruments in a similar fashion. An analytical calculation for both the Black-Scholes prices and Greeks can be found in "BSBarrierPricerAndGreeksAnalytical.m". Solutions for both vanilla options and all types of barrier options are implemented as provided by Wystup[2002].

The price of one DOBC is 4.5USD, the sensitivities to the greeks are as follows: $\delta = 0.5102, \gamma = 0.0304, \theta = -1.0790, vega = 26.0924$. As the bank, by selling our PPPN we are short 10796 such DOBC's. Our total Greek exposure can be summarized as follows:

$$\begin{cases} Value = -(4.5 * 10796) = -48582 \\ \delta = -(0.5102 * 10796) = -5508.3 \\ \gamma = -(0.0304 * 10796) = -328.6543 \\ \theta = -(-1.0790 * 10796) = 11649 \\ vega = -(26.0924 * 10796) = -281690 \end{cases}$$

4.2 Delta hedging

Since we are short 5508.3 delta it is obvious that buying 5508 stocks with a δ of 1 would result in an (almost) delta neutral portfolio. The cost of buying 5580 stocks equals 281130 euro. To finance this transaction we use the 48582USD from the DOBC options that we sold and borrow an additional 232550 at the risk free rate. Note that the added bankaccount position creates an added negative θ exposure of 118.17. The properties of the hedged portfolio can be summarized as follows:

Exposure	DOBC	Stock	BankAccount	Total
Amount	-10796	5508	-232550	N/A
Value	-48582	281130	-232550	0
Delta	-5508.3	5508	0	0
Gamma	-328.65	0	0	-328.65
Theta	11649	0	-118.17	10468
Vega	-281690	0	0	-281690

4.3 Delta-Vega hedging

In addition to our delta exposure we also want to hedge our vega exposure. To accomplish this we add an additional vanilla option to our hedging portfolio. We choose an ATM call with a relatively large offsetting vega (and positive delta) as our second hedging instrument. The price is 2.29USD, time to maturity is 241 days and the strike is set at 51. Greeks are as follows: $\delta = 0.4939$, $\gamma = 0.0630$, $\theta = -1.5397$, $vega = 16.3879$. We put our data and constraints in a linear system in order to obtain a delta-vega neutral solution. We buy 17189 of the vanilla call options and sell 298 stocks for a total cost of $(17189 * 2.29 = 39363) - (298 * 51.04 = 152132) =$ -112769USD. We have $112769 + 48582 =$ 161351USD left to put on the bankaccount. Our final combined position looks like this:

Exposure	DOBC	ATM Call	Stock	BankAccount	Total
Amount	-10796	17189	-298	161351	N/A
Value	-48582	39363	-152132.68	161351	0
Delta	-5508.3	8488	-2980	0	0
Gamma	-328.65	1082	0	0	753.96
Theta	11649	-26466	0	819.88	-13997.71
Vega	-281690	281690	0	0	0

Alternatively, we could have added two vanilla options to the hedging portfolio and force the hedging portfolios' price to match the DOBC portfolio price as an additional constraint. As a result, no extra money should be borrowed or put on the bankaccount. When adding additional options to our portfolio we make sure to use a variety of maturities and strikes in order to avoid strong linear dependencys. The first option that we use is an OTM put that matures in 38 days and has a strike price of 48.5. The characteristics of this vanilla option are as follows: $price = 0.25, \delta = 0.17, \gamma = 0.0926, \theta = -3.4468, vega = 4.1687$. For the second hedging option we use an OTM call with 612 days to maturity and a strike price of 65. The characteristics of this instrument are as follows: $price = 0.4150, \delta = 0.0982, \gamma = 0.01791, \theta = -0.4420, vega = 11.3773$. When solving this system of equations we obtain the following results:

Exposure	DOBC	OTM put	OTM call	Stock	Total
Amount	-10796	-11162	28849	773	N/A
Value	-48582	-2846.38	11972	39456	0
Delta	-5508.3	1900.10	2835	773	0
Gamma	-328.65	-1034.01	517	0	-845
Theta	11649	38474.91	-12752	0	37371
Vega	-281690	-46532.48	328226	0	0

4.4 Delta-Vega-Gamma-Theta hedging

In this subsection we create a portfolio that is Delta Vega Gamma and Theta neutral. We use the same procedure as before but use more hedging instruments in order to cover the additional constraints. This time we are obliged to impose the total cost of the hedging portfolio to match the value of the original shorted portfolio. The reason for this is that any leftovers that are put on the bankaccount or shortages that must be borrowed would result in additional theta exposure (We already demonstrated this concept in the previous subsection). We use the stock and the three vanilla options from previous subsections as our hedigng instruments in addition to another OTM put with 122 days to maturity and a strike price of 45. The characteristics of this instrument are as follows: $price = 0.4050, \delta = -0.1312, \gamma = 0.0364, \theta = -1.95, vega = 6.2739$. The final result is summarized in the table below:

Exposure	DOBC	OTM Put 1	ATM Call	OTM Call	OTM Put 2	Stock	Total
Amount	-10796	-9795	9364	6968	14315	410	N/A
Value	-48582	-2498	21443	2891	5797	20948	0
Delta	-5508.3	1667	4624	684	-1878	410	0
Gamma	-328.65	-907	589	125	521	0	0
Theta	11649	33763	-14417	-3079	-27915	0	0
Vega	-281690	-40834	153455	79263	89809	0	0

A MarketData and Assumptions

The XLB price was fetched from finance.yahoo.com. The price S_0 was set around 51.04 on May 19th 2015. Optiondata was downloaded via the R quantmod library's: View "WriteOptionDataToCSV.R". The dividend rate information was gathered from www.dividend.com while the US yield curve information was found on treasury.gov. All relevant data that is used in the Matlab code was added to .xls files in the "optiondata" subdirectory. Also view figure 2. below:

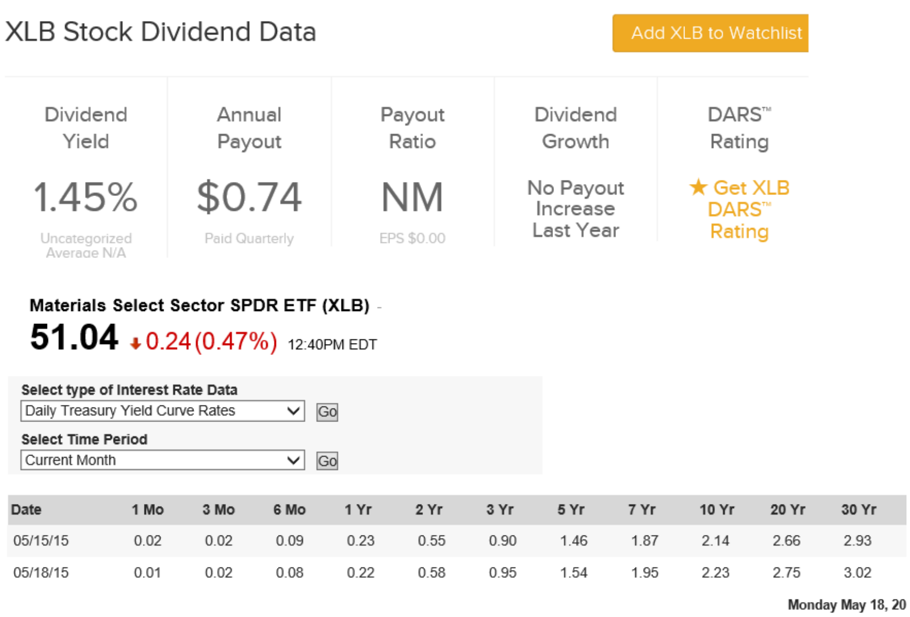

Figure 3: Sources of Marketdata

B Robustness Analysis Results

In the tables below the calibration and out of sample pricing results are illustrated for different in sample calibration techniques and weighting schemes. We show (adjusted) out of sample RMSE for Carr Madan pricing, straight vanilla monte carlo pricing and vanilla monte carlo pricing via randomized pricing of exotic barrier options ($Call = DOBC + DIBC$). Results are generated via "TestEarlyStoppingCrossValidation.m" and can be further discussed during the presentation.

Calibration	Weighting	Time	CM	MC-Vanilla	MC-Exotic
Full	Equal	29.92s	0.1284	0.1373	0.1266
Full	Bid-Ask	32.35	0.1322	0.1369	0.1344
Full	ImpliedVol	40.39s	0.1290	0.1321	0.1316
Stopping	Equal	0.56s	0.5205	0.4461	0.5185
Stopping	Bid-Ask	2.76s	0.3176	0.2598	0.3224
Stopping	ImpliedVol	1s	0.2775	0.2383	0.2808
CrossVal	Equal	10.98s	0.1289	0.1330	0.1272
CrossVal	Bid-Ask	11.3s	0.1367	0.1412	0.1380
Crossval	ImpliedVol	17.78s	0.1289	0.1315	0.1315

Table 1: Calibration Runtime and Out of Sample RMSE

Calibration	Weighting	Time	CM	MC-Vanilla	MC-Exotic
Full	Equal	29.92s	0.027	0.0285	0.0218
Full	Bid-Ask	32.35	0.0240	0.0244	0.0257
Full	ImpliedVol	40.39s	0.0216	0.0208	0.0226
Stopping	Equal	0.56s	0.4176	0.3436	0.4159
Stopping	Bid-Ask	2.76s	0.2124	0.1529	0.2168
Stopping	ImpliedVol	1s	0.1670	0.1327	0.1697
CrossVal	Equal	10.98s	0.0221	0.0246	0.0213
CrossVal	Bid-Ask	11.3s	0.0216	0.0269	0.0224
Crossval	ImpliedVol	17.78s	0.0215	0.0205	0.0225

Table 2: Calibration Runtime and Out of Sample Adjusted RMSE

C Monte Carlo Variance Reduction Results

In the table below we show the prices, runtime and 95% confidence intervals that are obtained when pricing the underlying DOBC option with and without monte carlo variance reduction techniques. We perform 150000 individual stockpath simulations for the non-antithetic runs while we generate 75000 pairs during the antithetic runs. Confidence intervals are calculated as 97.5% upper confidence level price minus the 97.5% lower confidence level price. Results are generated via "TestMonteCarloVarianceReductionAndStockSimulations" and can be further discussed during the presentation.

Property	No Var. Red.	AntiThetic	Control Variate	Combination
Price	4.51	4.51	4.51	4.51
nrPeriods	652	652	652	652
nrPaths	150000	75000	150000	75000
runtime	6.46s	3.67s	6.58s	3.65s
Conf. Interval	0.0633	0.0328	0.0363	0.0306