Algorithmic optimization and its application in finance

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht von

Kujtim Avdiu

18.08.2020

1. Gutachter/in: Prof. Stefan Mittnik, Ph.D.

2. Gutachter/in: PP Priv.Doz.Mag.Dr. Fabio Rumler

Tag der mündlichen Prüfung: 21.07.2021

Statutory declaration

I hereby declare that this thesis represents my own work, without the use of any unauthorized auxiliaries.

Eidesstattliche Versicherung

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Wien, 14.07.2021
Dipl.-Ing. Kujtim Avdiu

Acknowledgements

I would like to thank Professor Stefan Mittnik for his commitment to evaluate this work as well as Dr. Fabio Rumler for his supporting attitude.
I am especially thankful to Prof. Finsinger, who enabled the working environment for me at University of Vienna. The infrastructure was very supporting to my work and my computational demand.
I would also like to thank Prof. Stephan Unger for an excellent collaboration and assistance in discussion of the topics covered in this thesis.

Contents

Summary in German

Ziel dieser Arbeit ist es, verschiedene Fragestellungen aus dem Bereich der Finanzwirtschaft unter Anwendung finanzmathematischer Modelle und Optimierungsverfahren zu untersuchen.

Bevor ein Modell auf eine bestimmte Problemstellung angewendet werden kann, müssen zuerst die Modellergebnisse an die Beobachtungsdaten angepasst werden. Hierfür wird eine Zielfunktion definiert, die mit Hilfe von Optimierungsalgorithmen minimiert wird. Damit können die optimalen Modelparameter gefunden werden. Dieses Verfahren nennt sich Modellkalibrierung oder Modellanpassung und setzt voraus, dass das Modell für dieses Anwendungsgebiet geeignet ist.

In dieser Arbeit kommen finanzmathematische Modelle, wie Heston, CIR oder geometrische Brownsche Bewegung und statistische Methoden, wie Inverse-Transformation und Chi-Quadrat-Test zur Anwendung. Darüber hinaus werden folgende Optimierungsmethoden getestet: Genetischer Algorithmus, Particle-Swarm-, Levenberg-Marquardt- und Simplex-Verfahren.

Der erste Teil dieser Arbeit beschäftigt sich mit der Problematik, einen genaueren Prognoseansatz für Marktliquidität zu finden, in dem man anstelle der Standard-Brownschen-Bewegung ein kalibriertes Heston-Modell für die Simulation der Bid-/Ask-Pfade und die Inverse Transformationsmethode statt Compound-Poisson-Prozesses für die Generierung der Bid-/Ask-Volumsverteilungen verwendet. Es kann dabei gezeigt werden, dass die simulierten Handelsvolumina zu einem einzigen Wert konvergieren. Dieser Wert kann als Marktliquiditätsschätzer verwendet werden, wobei das Heston Modell dafür besser geeignet ist als die Standard-Brownsche-Bewegung. Gleichzeitig erzielt die inverse Transformationsmethode bessere Ergebnisse für die Simulation der Bid/Ask-Volumen.

Im zweiten Teil untersuchen wir den Preisaufschlag für Hedging- bzw. Liquiditätskosten, den Kunden zahlen müssen, wenn sie strukturierte Produkte kaufen. Dafür replizieren wir den Payoff von zehn verschiedenen strukturierten Produkten und vergleichen ihren fairen Preisen mit den tatsächlich gehandelten Preisen. Für diesen Zweck kommt paralleles Rechnen zum Einsatz, eine neue Technologie, die in der Vergangenheit nicht möglich war. So können wir ein kalibriertes Heston-Modell für die Berechnung des fairen Preises von strukturierten Produkten über einen längeren Zeitraum anwenden. Unsere Ergebnisse zeigen, dass der Aufschlag, den die Kunden für diese zehn Produkte zahlen müssen, zwischen 0,9%-2,9% liegt. Ebenfalls können wir beobachten, dass Produkte mit höheren Payoff-Levels, bzw. besseren Kapitalschutz, höhere Kosten erfordern. Weiters identifizieren wir Marktvolatilität als statistisch signifikanten Treiber des Preisaufschlags.

Im dritten Teil zeigen wir, dass der Tracking-Error eines passiv gemanagten ETFs durch den Einsatz von Optimierungsmethoden signifikant reduziert wer-

den kann, wenn der Korrelationsfaktor zwischen Index und ETF in der Zielfunktion berücksichtigt wird. Durch das Finden von optimalen Gewichten von einem selbst-konstruierten Anleihen-Index und des DAX-Index, kann die Anzahl der Konstituenten stark reduziert werden, während der Tracking-Error ebenfalls signifikant reduziert werden kann.

Im vierten Teil entwickeln wir eine Hedgingstrategie basierend auf Treibstoffpreisen, die vor allem für Endbenutzer von Benzin- und Dieselkraftstoffen Anwendung finden kann. Das ermöglicht dem Treibstoffkonsumenten durch den Erwerb von Kaufoptionen, Treibstoff zu einem bestimmten Preis für eine bestimmte Zeit zu kaufen. Zum Pricing der amerikanischen Kaufoption verwenden wir eine geometrisch-Brownsche Bewegung, kombiniert mit einem binomischen Modell.

Summary in English

The goal of this thesis is to examine different issues in the area of finance and application of financial and mathematical models under consideration of optimization methods.

Prior to the application of a model to its scope, the model results have to be adjusted according to the observed data. For this reason a target function is defined which is being minimized by using optimization algorithms. This allows finding the optimal model parameters. This procedure is called model calibration or model fitting and requires a suitable model for this application.

In this thesis we apply financial and mathematical models such as Heston, CIR, geometric Brownian motion, as well as inverse transform sampling, and Chi-square test. Moreover, we test the following optimization methods: Genetic algorithms, Particle-Swarm, Levenberg-Marquardt, and Simplex algorithm.

The first part of this thesis deals with the problem of finding a more accurate forecasting approach for market liquidity by using a calibrated Heston model for the simulation of the bid/ask paths instead of the standard Brownian motion and the inverse transformation method instead of compound Poisson process for the generation of the bid/ask volume distributions. We show that the simulated trading volumes converge to one single value which can be used as a liquidity estimator and we find that the calibrated Heston model as well as the inverse transform sampling are superior concerning the use of the standard Brownian motion, resp. compound Poisson process.

In the second part, we examine the price markup for hedging or liquidity costs, that customers have to pay when they buy structured products by replicating the payoff of ten different structured products and comparing their fair values with the prices actually traded. For this purpose we use parallel computing, a new technology that was not possible in the past. This allows us to use a calibrated Heston model

to calculate the fair values of structured products over a longer period of time. Our results show that the markup that clients pay for these ten products ranges from 0.9%-2.9%. We can also observe that products with higher payoff levels, or better capital protection, require higher costs. We also identify market volatility as a statistically significant driver of the markup.

In the third part, we show that the tracking error of an passively managed ETF can be significantly reduced through the use of optimization methods if the correlation factor between Index and ETF is used as target function. By finding optimal weights of a self-constructed bond- and the DAX- index, the number of constituents can be reduced significantly, while keeping the tracking error small.

In the fourth part, we develop a hedging strategy based on fuel prices that can be applied primarily to the end users of petrol and diesel fuels. This enables the fuel consumer to buy fuel at a certain price for a certain period of time by purchasing a call option. To price the American call option we use a geometric Brownian motion combined with a binomial model.

1 Introduction & Methodology

1.1 Introduction

The aim of the work is to investigate various optimization problems that arise in the field of finance. In order to solve these optimization problems, a target function is defined and minimized by applying the optimization methods. The optimal parameters are then used for modelling. In this thesis the following optimization methods are applied: Genetic algorithm, Particle Swarm Optimization, Levenberg-Marquardt method and Simplex method. Financial mathematical models, such as Heston, CIR or geometric Brownian motion, as well as statistical methods such as Inverse transform sampling and Chi-square test are also applied. Because of the high computing time, most investigations have been implemented on cluster systems, a new technology using parallel computing, which was not possible in the past due to the unavailability of the technology.

The first paper deals with the estimation of market liquidity based on the simulation of bid ask prices and bid ask volumes. Market liquidity is assumed to be a function of price and volume. By simulation of the both, bid/ask prices as well as corresponding bid/ask volumes, we check for every crossing time where the bid price is equal or greater than the ask price. This is an indication that a buyer and a seller have agreed on a price at which they want to exchange the minimum amount out of the offered and demanded quantity. This procedure is repeated for every intersection point of the bid and ask prices along the simulated price path. Following standard numerical option pricing procedure, by multiple simulation of the price paths and calculation of the average of all synthetically generated trades, the expected traded volume given a certain time interval, converges to one equilibrium, which can be interpreted as the implied market liquidity. This incorporates the bid ask spread price dynamics as well as the underlying market depth.

We then try to identify a superior forecasting method for this market liquidity measure, using a calibrated Heston model for the bid/ask price path simulation, instead of standard Brownian motion and compound Poisson process, namely the inverse transform sampling for the generation of the bid/ask volume distribution. The goal is to apply these techniques to market liquidity modeling and identify any superior forecasting quality to the liquidity estimation and test this approach with high frequency data like DAX- and ESTX50- Future. The outstanding new idea within this work is the application of various optimization techniques to the Heston model in order to analyze its forecasting quality in contrast to the standard Brownian motion.

Heston's stochastic volatility model [7] seems to be a better choice when modeling price fluctuations, even though it makes assumptions which might not be encoun-

tered in reality such as normal distributed returns in short time frames. The application of stochastic volatility models is very broad. In this paper we will specifically investigate the application of the Heston model to market liquidity simulation.

Market liquidity describes the ability of a financial market to absorb additional trades between market participants without affecting the price. When price impact is strong we speak of illiquidity of a market. In order to simulate market liquidity a certain parameter set is necessary to map trade capacity. This includes quoted bid and ask prices along with quoted bid and ask volume, as well as traded price and traded volume.

The basis of this research within this chapter is the paper of Unger and Hughston [8] which investigates the liquidity generating price process. It incorporates two standard Brownian motions which represent the price process of the buyer and the seller. The intersection of the two Brownian motions serve as an indicator for the corresponding volume generating process and therefore as an indicator variable for the points in time when liquidity is being generated. A simulated trade occurs by taking the minimum of two quoted sizes. Since the quoted volume is assumed to be simulated by a Compound Poisson process, the law of large numbers yields exponentially distributed random numbers. The interesting question that arises will be how the convergence process according to the law of large numbers looks like if we take the minimum of two exponentially distributed random variables.

My hypothesis suggests that applying the Heston model resp. inverse transformation method for market liquidity estimation might be a better choice than using the standard geometric Brownian motion resp. compound Poisson process.

We apply the liquidity estimator to high-frequency data and find a highly significant prediction power of market liquidity on two major European futures, the DAX and ESTX50 future and show that the simulated trading volumes converge to one single value, which can be used as liquidity estimator. After this application we observe that the calibrated Heston model as well as the inverse transform sampling are superior when comparing to the use of standard Brownian motion, resp. compound Poisson process.

In the second work we replicate the payoff of ten different structured products and compare their calculated fair values with the real traded prices in order to determine the markup incurred in the products. This is the first time that a calibrated Heston model is used to replicate the payoffs of a single asset structured products. We find significant markups, ranging from 0.9%-2.9% markup on the fair value, which implies that the better the payoff of the product for the investor, the higher the hedging costs for the bank. Based on these results two possible drivers of markups are tested: Volatility and market liquidity. For 8 out of 10 products, volatility is being obtained as a significant driver of the markup, while liquidity is only found to be a driver of the markup in one product.

In the third paper of the thesis we show that the tracking error of passively managed ETFs can be significantly reduced by certain optimization techniques, when the correlation factor is considered in the target function. The various optimization methods for simulated as well as real traded prices of bond indices are applied in order to find the optimal weights for a constructed bond-ETF. By minimizing the tracking error while keeping the number of bonds small, which then minimizes the allocation costs, we find that the tracking error can be also minimized.

To simulate the bond prices with 10 years maturity and different credit spreads we need to simulate the short rate with calibrated Cox-Ingersoll-Ross (CIR) model.

Then we construct the ETF by a combination of bonds with different weights. In order to find the optimal weights and minimize the tracking error we use different optimization techniques and show that a performance approximation of a bond index by a bond ETF is possible.

The main goal of bond index tracking is to keep the number of bonds small, while minimizing the allocation costs, which in turn keeps the tracking error at a predetermined minimum level. Finally we analyze the results and compare the optimization methods based on the tracking error for simulated as well as realized bond index prices.

The fourth paper applies geometric Brownian Motion price simulation to fuel prices in order to price American-style fuel price Call options, where the risk-free change in fuel price is assumed to follow the inflation rate. The purpose of such an option is to introduce the hedging possibility for a consumer at the gas station. We highlight the potential application of a fuel option for companies and individuals who are highly exposed to road traffic-based vehicles such as cars and trucks, as there does not exist a hedging possibility for refined end products for non-institutional clients, starting from lack of products to minimum size requirements for trade eligibility of certain crude financial instruments such as crude oil futures.

1.2 Methodology

Market Liquidity estimation in a high-frequency setup

The measurement of market liquidity is important for several aspects. Firstly, the tradability of an asset price and its hedging depends on the market liquidity. Big positions covered by banks are subject to constant market liquidity revision when it comes to unwinding the latter. By calculation of an expected market liquidity value it is possible to detect liquid trading windows and minimize slippage when a large volume needs to get un-winded within a certain time window. Furthermore, mutual funds can use the liquidity measure when they expect to receive certain amount of capital. In order to be able to ensure certain performance the market entry is crucial. Therefore an estimation of future market liquidity helps to improve mutual fund performance.

Unger and Hughston [8] measure liquidity using a numerical simulation methodology similar to the one which is used in Monte Carlo option pricing. My idea is to expand the methodology for simulating the underlying price process by application of the Heston model, as well as expanding the methodology for simulating the underlying volume process by applying the inverse transform sampling. The novelty about this expansion is that this is the first time that the Heston model is being applied to the simulation of market liquidity. At the same time, it appears to be the first time that the inverse transform is being applied to the simulation of an order volume generating process. In order to measure the forecasting accuracy, I also conduct forecasting error measurements by comparing the quality of the new forecasting methods to the old ones.

The literature addresses the application of the Heston model to risk estimation and hedging, as well as to the pricing of derivatives and structured products. The computational integration of such methods is covered by many research works such as by Ingber [3] or Kirkpatrick [12] while theory on optimization techniques is presented by research works as in Mikhailov and Nögel [5] or Gatheral [6]. However, there exists currently no research work that applies the Heston model to market liquidity measurement.

The paper of Unger and Avdiu [2] deals with the theoretic idea of making market liquidity tradable. In that sense, the displayed value of future expected market liquidity could define the terms of trade between two counterparts who are willing to offset each other's opinion on expected traded volume. Since this value could be traded like a contract for difference (CFD), this paper focuses on the hedging of such positions if there is no counterpart to offset a trader's position.

In order to measure market liquidity it is important to simulate the quoted bid and ask prices, as well as the quoted bid and ask volumes. Therefore the estimation of market liquidity assumes two stochastic processes for the development of the bid

(i) and ask (j) prices, two compound stochastic processes for the simulation of bid (i) and ask (j) volumes and an algorithm for the calculation of the arithmetic mean of the traded volumes over a certain period of time.

We calculate the arithmetic mean by taking taking the sum of all generated traded volumes per time interval and divide it by the total number of the time steps of the simulation. The bid/ask price process is assumed to follow a geometric Brownian Motion as in [9]. This enables us to compare it with the performance of the Heston model simulation.

We develop the multivariate geometric Brownian motions for the price path development of a buyer S_j and of a the seller S_j, with $\mu_{i/j}$(bid/ask price drift) and $\sigma_{i/j}$(price volatility), through:

$$S_i(t) = S_i(0)\, e^{\mu_i t + \sigma_i X_i} \tag{1}$$

$$S_j(t) = S_j(0)\, e^{\mu_j t + \sigma_j X_j} \tag{2}$$

where the correlation matrix between the Wiener processes X_i and X_j is defined as $E(X_i, X_j) = \rho_{i,j}$ and $\rho_{i,i} = 1$.

The Heston model assumes a stochastic volatility development of the bid (i) and ask (j) prices ($S_{i,j}$) with parameters $\mu_{i,j}$(bid/ask price drift), $V(t)_{i,j}$(bid/ask price variance), κ (rate of mean reversion), $\omega_{i,j}$(long run variance), $\sigma_{i,j}$(volatility of variance) and $W_{1,2}^{i,j}$ (Standard Brownian movements).

$$\frac{dS(t)_i}{S(t)_i} = \mu_i dt + \sqrt{V(t)_i} dW_1^i, \tag{3}$$

$$dV_i = \kappa_i(\omega_i - \sigma_i)dt + \sigma(t)_i \sqrt{V(t)_i} dW_2^i \tag{4}$$

describes the bid price development, while

$$\frac{dS(t)_j}{S(t)_j} = \mu_j dt + \sqrt{V_j} dW_1^j, \tag{5}$$

$$dV_j = \kappa_j(\omega_j - \sigma_j)dt + \sigma(t)_j \sqrt{V(t)_j} dW_2^j \tag{6}$$

describes the ask price development. W_1 and W_2 are correlated through $dW_1 \cdot dW_2 = \rho dt$.

In order to estimate the expected market liquidity we simulate independently bid and ask volumes by using compound Poisson process as well as inverse transform sampling. By performing a Chi-square test we evaluate the forecasting performance of both methods and identify the superior one. We assume that the dynamics of

the volumes processes Q_i and Q_j follow a compound Poisson process with jump rate λ. Therefore we define for the the bid and ask volume developments the following compound Poisson processes:

$$P(Q_i(t) = n) = e^{-\lambda_i t}\frac{(\lambda_i t)^n}{n!}, \tag{7}$$

$$P(Q_j(t) = n) = e^{-\lambda_j t}\frac{(\lambda_j t)^n}{n!}, \tag{8}$$

where Q_i is the quoted bid volume at time t and Q_j the quoted ask volume at time t.

Using the inverse transform sampling method, the volume process developments of Q_i and Q_j are assumed as follows. Let

$$P(X_i = x_i^k) = p_i^k, k = 1, 2, ...n_i, with \sum_{k=1}^{n_i} p_i^k = 1 \tag{9}$$

$$P(X_j = x_j^k) = p_j^k, k = 1, 2, ...n_j, with \sum_{k=1}^{n_j} p_j^k = 1 \tag{10}$$

be the probability mass function (PMF) of of bid (i) resp. ask (j) volume calculated by historical data with n_i quoted bid volumes $x_i^k, k = 1...n_i$ and n_j quoted ask volumes $x_j^k, k = 1...n_j$. The PMF of quoted bid (i) resp. ask (j) volume at time t is then defined by

$$P(Q_i(t) = x_i^k) = P(F(x_i^{k-1}) \leq U \leq F(x_i^k)) \tag{11}$$

$$P(Q_j(t) = x_j^k) = P(F(x_j^{k-1}) \leq U \leq F(x_j^k)) \tag{12}$$

with $F(x_{i/j}^t)$ being the cumulative distribution function (CDF) of historical quoted bid (i) and quoted ask (j) volumes at time t calculated from the PMF

$$F(x_{i/j}^k) = \sum_{t=1}^{k} p_{i/j}^t, \tag{13}$$

and U being a uniform random variable between 0 and 1.

We will further prove that inverse transform sampling is be a useful estimation method for the bid (i) and ask (j) volume process generation:

$$P(Q_{i/j}(t) = x_{i/j}^k) = \sum_{t=1}^{k} p_{i/j}^t - \sum_{t=1}^{k-1} p_{i/j}^t = p_{i/j}^k. \tag{14}$$

The result is a compound volume process $Y(t)$ at time t which characterizes the volume developing process.

$$Y(t) = \begin{cases} min\{Q_i(t), Q_j(t)\} & \text{if } S_i(t) \geq S_j(t) \\ 0 & o.w. \end{cases} . \tag{15}$$

As we can see, we always take the minimum of two quoted bid and ask volumes if the simulated bid price is at or above the simulated ask price, since this volume is assumed to be the agreed quantity of the transaction. Therefore, this process develops a path of simulated traded volumes.

$$E[Y] = \frac{\sum_{t=1}^{n} Y(t)}{n}. \tag{16}$$

Since the Heston parameters need calibration, we apply four different optimization techniques and compare their estimation quality in-sample and out-of-sample and test it on Euro Stoxx 50 Future options. The optimization methods we used for calibration are Genetic Algorithms (GA), the Particle Swarm Optimization technique (PSO), the Levenberg-Marquardt method (LM) and the Nelder-Mead Simplex method (SM). In addition to the four optimization methods, a combination of Levenberg-Marquardt and Genetic (LM +GA) algorithms is tested, using the optimal parameters of LM as starting values for GA.

These optimization methods lead to a minimization of the root mean square error between the estimated plain vanilla Heston option price and the realized option price of option i:

$$arg\ min_{\Omega} \sqrt{\frac{\sum_{i=1}^{N} (C_0(i, r, M_i, S, K_i) - C_i)^2}{N}}, \tag{17}$$

where Ω is the set of Heston parameters to be estimated, N expresses the number of options on the estimation day, C_0 is the Heston call function denoting the dollar adjusted call plain vanilla option price, r the interest rate, M_i the Maturity of Option i, S the closing price of the Underlying and K_i the strike of option i.

The order submission flow on the bid and ask volume is simulated by compounding Poisson process and inverse transform sampling. Both methods are assumed to reflect the real traded order arrival times. Therefore we perform a Chi-Square optimization test in-sample as well as out-of sample. Our null-hypothesis states that the real arrival times of conducted trades follows a compound Poisson distribution or an inverse transformation.

To estimate the Heston parameters, we use the same strat values, the same lower and upper bounds, and the same termination conditions for all four optimization methods. These are the optimal starting parameters as calculated from the historical bid/ask call data. The indicated optimization parameters are tested on the basis of all traded bid and ask options on 20.08.2015.

Table 1: Lower bounds, upper bounds and starting values of parameter estimation.

	kappa	theta	sigma	rho	v0
Lower Bounds	0.0001	0.0001	0.0001	-1	0.0001
Upper Bounds	100	10	10	1	10
Start	10.9	0.01	0.3	0.3	0.04

Table 2: Termination conditions

Maximum Function Evaluation	1,000
Maximum Iterations	20,000
Termination Tolerance on the Function Value	0.001

The overall test results show the smallest RMSE for the LM method, indicating that LM is superior to the other optimization methods. LM not only produces the smallest RMSE, but also is the fastest optimization method in terms of computational time. Although LM produces the best calibration results among all optimization methods, we can achieve improvement in test results by combining LM method with Genetic Algorithms.

Table 3 shows the RMSE error for all 4 optimization methods.

Table 3: Estimation RMSEs of all options 27.04.2015-30.12.2015

	GA	PSO	LM	NMSim	LM+GA
In Sample	46.85	15.30	11.27	11.85	10.23
Out of Sample	82.78	29.51	19.25	22.45	17.36
Computational Time in Seconds	38.106	61.248	2.610	5.742	42.987

To evaluate the sampling of bid/ask volumes by the chi-square test, we take historical FESX50 tick data and accumulate all tick changes, regardless of price or volume changes or the submission of new orders, at each second. We use tick data from April 26, 2015 to December 30, 2015 and test both in-sample and out-of-sample.The in-sample test includes 1-50 days, while the out-of-sample test takes these 1-50 in-sample days as the estimation set and tests 1 day out-of-sample.

Our results show that the inverse transformation method with a depth of 1 day of historical data is more suitable than the composite Poisson process for sampling bid/ask volumes. These results hold for both in-sample and out-of-sample data.

This paper compares the estimation performance of the Heston model and the Geometric Brownian Motion (GBM) model (bid-ask prices) as well as the Inverse Transformation Sampling and Compound Poisson processes (bid-ask volumes). The application is a unique approach to estimating market liquidity.

Our results show that by simulating bid and ask prices generated by the calibrated Heston model, market liquidity can be estimated up to 29.14% better than with the GBM model. The results are robust for both in-sample and out-of-sample tests. The only drawback of using the Heston model is the high computational time required to calibrate the parameters and simulate the prices.

Implicit Hedging and Liquidity Costs of Structured Products

Structured products allow a different type of investment and are linked to certain conditions. The price of a structured product is thus derived from the development of the underlying and the predetermined conditions. Banks issuing such products must at the same time consider how to hedge and trade corresponding counter positions, since they face the requirement to hedge instantly when the order of a client arrives and the position is opened. Structured products are based on complex mathematical formulas and can incorporate high risk exposures. The fair value of these products calculated by using financial models can differ from the real observed prices on the market. This difference is called markup, which is also subject to change over time. The drivers of this markup and the change in the markup have not been extensively studied by the literature.

Henderson and Pearson(2011) [4], Stoimenov and Wilkens(2005) [10], Bergstresser(2008) [11], Rogalski and Seward(1991) [15], and Jarrow and O'Hara(1989) [16] take the closing prices of linked structured products and compare them to theoretical values derived from the prices of options traded on the underlying of those products. As they find that the real traded prices are greater than the calculated fair market values obtained using option pricing methods, they lack on identifying the real driver behind the discrepancy between pricing and real observed prices.

Due to the lack of available technologies, it has not been possible in the past to use financial mathematical models such as Heston to determine the fair value of structured products over a long period of time.

Therefore we develop a framework which uses the Heston model, including the calibration methods studied above, resp. geometric Brownian Motion in order to replicate the payoffs of ten structured products and measure the spread between theoretical and real prices.

We calculate the payoffs of 10 structured products according to the terms which appear on the fact sheets published by the banks. Some of the products contain multiple underlyings. In order to account for that, we use a multivariate geometric Brownian motion which allows us to calculate a more accurate fair value:

$$dS_t^i = X_t^i dt + \sigma_i S_t^i dW_t^i, \tag{18}$$

where the Wiener processes are correlated such that $\mathbb{E}(dW_t^i, dW_t^j) = \rho_{i,j}, dt$, where $\rho_{i,i} = 1$.

Product payoffs which only consider one underlying, S_t are priced using the Heston model:

$$dS_t = \mu S_t dt + \sqrt{\nu_t} t S_t dW_t^S, \tag{19}$$

where ν_t, the instantaneous variance, is a CIR process:

$$d\nu_t = \kappa(\theta - \nu_t)\, dt + \xi\sqrt{\nu_t}, dW_t^{\nu}, \tag{20}$$

and W_t^S, W_t^{ν} are Wiener processes (i.e., random walks) with correlation ρ, or equivalently, with covariance ρ dt, where μ is the rate of return of the asset, θ is the long run variance, κ is the rate at which ν_t reverts to θ, ξ is the volatility of the volatility and determines the variance of ν_t.

For calibration of the Heston parameters, we use the Levenberg Marquardt method and Genetic algorithms, since this combination turns out to be the best method in order to minimize the root mean square error between the estimated plain vanilla Heston option price and the realized option price of option i:

$$arg\ min_{\Omega} \sqrt{\frac{\sum_{i=1}^{N}(C_0(i, r, M_i, S, K_i) - C_i)^2}{N}}, \tag{21}$$

where Ω is the set of Heston parameters to be estimated, N the number of options on the estimation day, C_0 is the Heston call function denoting the dollar adjusted call plain vanilla option price, r the interest rate, M_i the Maturity of Option i, S the closing price of the Underlying, K_i the strike of option i. Each simulation generates 500,000 random paths for each underlying, to which all ten payoffs of the structured products are then applied to. The fair value is then calculated by taking the arithmetic average among all outcomes.

The optimization results obtained for the Heston model are then used to calculate the fair value of structured products linked to a single asset.

In order to identify the drivers of the markups we run two regression, controlling for volatility and liquidity in order to test if they explain the markup. For structured products which include multiple underlyings, we run a multivariate regression for $Markup_{i,t}$ for product i at time t, on the volatility $\sigma_{j,t}$ of underlying j:

$$Markup_{i,t} = \alpha + \sum_{j=1}^{N} \gamma_j \sigma_{j,t} + \epsilon_{i,t}. \tag{22}$$

as well as on liquidity $\lambda_{j,t}$ of product i of underlying j:

$$Markup_{i,t} = \alpha + \sum_{j=1}^{N} \gamma_j \lambda_{j,t} + \epsilon_{i,t}. \tag{23}$$

For products based on just one underlying we perform a standard single regression, controlling for volatility $\sigma_{i,t}$ of product i:

$$Markup_{i,t} = \alpha + \gamma_p \sigma_{i,t} + \epsilon_{i,t} \tag{24}$$

as well as for liquidity $\lambda_{i,t}$ over time:

$$Markup_{i,t} = \alpha + \gamma_p \lambda_{i,t} + \epsilon_{i,t}. \tag{25}$$

The 10 structured products were replicated based on their payoffs and priced using multivariate geometric Brownian motion or the optimal Heston model (fair value). The difference between fair value and historical prices is defined as hedging and liquidity costs (markup).We find that the average markup for a structured product ranges from 0.9% to 2.9%.

Table 4: Hedging and Liquidity Costs

Product	Min	Max	Average
Dual Index kick-out, Capital risk	-0.7%	3.8%	2.0%
Dual Index + Coupon, Capital risk	0.1%	1.4%	0.9%
Trippel Index + Coupon, Capital risk	-0.5%	1.4%	1.1%
Single Index + Coupon, Capital risk	-0.2%	1.6%	1.4%
Single Index + maturity Coupon, Capital protection	-0.3%	1.8%	1.3%
Single Index + Payoff, Strike Capital protection	-0.7%	3.5%	2.1%
Single Index + Barrier Payoff, Capital protection	0.3%	2.9%	1.4%
Single Index + ongoing Payoff, Capital protection	0.5%	2.9%	1.9%
Single Index + Payoff, Capital protection	0.6%	4.1%	2.2%
Single Index, Capital risk	1.3%	3.1%	2.9%

We are the first to use a calibrated Heston model to replicate the payoffs of single asset structured products. The replication is done by using server calculations, a new technology that uses parallel calculations, which was not possible in the past due to the unavailability of the technology.

Based on these results, we are able to identify possible drivers of this markup, specifically whether these drivers explain the change in markup over time. We test two possible drivers: volatility and market liquidity of the underlying. We find significant results for volatility for 8 out of 10 products, while we identify liquidity as a driver for only one product, where the payoff of this product does not require hedging.

Our results suggest that most of the premium can be attributed to hedging costs, while hedging costs due to market liquidity cannot be extracted for complex payoffs, as the volatility of the underlying seems to dominate the cost of hedging in complex structures. In the presence of hedging, liquidity costs are embedded in the hedging costs of the underlying.

Minimizing the Index Tracking Error using Optimization Methods

Another application of optimization techniques in the world of finance is being presented by minimizing the index tracking error of bond ETFs.

The purpose of ETFs is to replicate the performance of an underlying index. Passive portfolio managers seek to minimize the discrepancy between the net asset value (NAV) of an ETF and its underlying index by buying and selling the underlying assets accordingly.

Since each transaction is associated with transaction costs, the managers face the trade off between tracking the underlying index as good as possible while maintaining the lowest possible transaction costs. Blanchett and Blanchett(2007) [13] give an overview of the cost drivers of these transactions.

Jeurissen and van den Berg [14] found that index tracking can be significantly optimized by using hybrid genetic algorithms to minimize tracking error. This method showed better performance than a randomly selected portfolio used to track the performance of the AEX index.

The methodology of our approach to minimize the index tracking error consists on the novel approach using the correlation factor between index and ETF $\rho_{(idx,ETF)} \rightarrow 1$ as a target function. Then the quality and robustness of the four optimization methods is compared and applied to the tracking of a virtually constructed fixed income bond ETF. The constructed bond index consists of 10 bonds, each with maturity 10 years while based on different credit spreads, using current Euro Swap rates or market data. Then an ETF which exhibits a minimal tracking error is being replicated. The goal is to forecast the price development of the ETF as good as possible by conducting a one-time asset allocation in advance.

Finally, an out-of-sample test with 10,000 simulations using the four different optimization techniques is being performed in order to identify the best optimization method. The pricing model assumes to follow a Cox-Ingersoll-Ross (CIR) model with a short-rate, r_t, satisfies

$$dr_t = \alpha(\mu - r_t)dt + \sigma\sqrt{r_t}dW_t,\ \alpha, \mu, \sigma > 0, r \geq 0, \tag{26}$$

where W_t is a standard Brownian motion, α is the speed of adjustment, μ is the long term average rate (the mean-reverting level) and $\sigma\sqrt{r_t}$ is the implied volatility. The zero-coupon bond prices are calculated by

$$P_t^\tau = E_t^*[exp(-\int_t^{t+\tau} r_u du)], \tag{27}$$

where E^* denotes expectation under the risk-neutral probability measure Q^*.

Then we construct a bond index at time t, consisting of 10 bonds, each of maturity 10 years for 1250 trading days which is calculated by

$$Index_t = Index_{t-1} \sum_{n=1}^{10} w_n \frac{B_n(t)}{B_n(t-1)}, \tag{28}$$

where w_n is the n-th weight of the corresponding bond and $B_n(t)$ is the n-th bond price at time t with maturity 10 years and $Index_1 = 100$. Then we simulate 10 bond price paths with randomly chosen vector of credit spreads from 2% to 3% and compare it to a real traded bond index which was constructed, using 10 different index bonds with different maturities.

For the construction of the ETF based on market data, ten zero-coupon bonds issued by different banks are used.

The weight vector for the market and simulated data is randomly chosen with $w = (0.2, 0.15, 0.1, 0.07, 0.09, 0.12, 0.15, 0.03, 0.02, 0.07)$.

The ETF is then constructed by allocating different combination of bonds with different weights. In order to find the optimal allocation the four optimization methods are applied for constructing the ETF. The methods are mentioned above. We test for an interval of $[4; 7]$ bonds. For each allocation we run the four different optimization methods and evaluate its tracking error.

Since we want to calculate the tracking Error of the bond ETF we download daily net asset values of the bond index and calculate their returns. Then we take the allocation weights for the next 250 trading days and bring them in matrix form, consisting of two matrices, X and Y:

$$X = \begin{bmatrix} R_{ETF_i,1001} \\ R_{ETF_i,1002} \\ R_{ETF_i,1003} \\ \cdot \\ \cdot \\ R_{ETF_i,1250} \end{bmatrix} \quad Y = \begin{bmatrix} R_{Index,1001} \\ R_{Index,1002} \\ R_{Index,1003} \\ \cdot \\ \cdot \\ R_{Index,1250}, \end{bmatrix}$$

where $R_{ETF,t}$ is the return on the ETF-portfolio at time t, which contains different bonds. $R_{Index,t}$ is the return on the NAV of the forecasted index benchmark at time t.
The tracking error is defined through the standard deviation of the active returns, which is the difference between NAV returns and ETF-portfolio returns. We calculate it by

$$TrackingError = \sqrt{Var(X - Y)} * \sqrt{250}. \tag{29}$$

After a time window of 1.000 trading days we calculate the optimal bond weights. These weights are used for tracking the index, where the tracking error is minimized. Then we take these weights and apply them to the next year. In other words, the weights are kept constant for the last 250 trading days in order to perform an out-of-sample test. The goal is to maximize the correlation between the return of the index price arising from our asset allocation and the realized ETF price:

$$arg\ max_{\Omega}(\rho_{index,ETF}), \tag{30}$$

where Ω is the set of ETF weights to be estimated. Since we want to replicate the bond index by using a new approach we use the correlation between the index and the ETF:

$$\rho_{index,ETF} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{31}$$

where x_i is the i-th return of the projected bond index and y_i is the i-th return of the tracking ETF.

In this paper, we propose a new approach to minimize index tracking error by setting the correlation factor between index and ETF $\rho_{(idx,ETF)}$ to $1 as an objective function$.

We then compare the quality and robustness of four different optimization methods applied to the tracking of a stock ETF index and two virtually constructed bond ETFs. For this purpose, we use 3 different benchmarks: simulated bond ETF, real data ETF and DAX ETF. We then perform an out-of-sample test using four different optimization techniques, such as Genetic Algorithms, Particle Swarm Optimization, Levenberg-Marquardt, and Nelder-Mead Simplex.

As a last step we assign the target function to each optimization method. This allows us to compare the degree at which the tracking error could be minimized. We perform this procedure for over 5 years with different bond allocations.

In summary, we find that the Levenberg-Marquardt algorithm works best for ETF tracking for both simulated and real data, and its application to the objective function pushing the correlation between the index and the ETF towards 1.

Fuel hedging in an inflated environment

Another example of price path simulation to the real world application is presented when pricing an American call option on fuel, where the risk-free change in fuel price is assumed to follow the inflation rate.

Commercial customers and retail clients would benefit from the possibility of hedging themselves against increases in fuel prices. Ideally they would be able to buy such protections directly at the gas station when they fuel their cars, trucks, etc. Currently there does not exist the possibility for retail clients to hedge the refined end product price. Since such products are mainly used and available, e.g. due to minimum size requirements, etc. to institutional clients, this would open not only a whole new market and business opportunity for providers of such insurers, but also enable retail clients to benefit from small contract sizes which are directly linked to their net exposure.

Fuel demand is highly correlated to economic growth. Therefore, the general view is that fuel prices will tend to increase over time. There exist certain underlying conditions which back this claim.

On the one hand, countries like China and India have an enormous growth potential, and will therefore require a lot of fuel in order to satisfy domestic demand. Since renewable energy resources will still take a while until they are widely used, fossil-based products will continue to serve as primary source of energy supply in the short term. This means that in a short- to mid-term period, oil and fuel prices are likely to increase.

On the other hand, the price vulnerability of the oil price due to geo-political tensions in the Middle-East, the region which is the second-largest provider of the global oil supply, is the main cause for high risk exposure of oil consumers. Due to the increased and unstable volatility because of the dependency on political developments on these countries, oil consumers are in desperate need of hedging instruments which cover this risk exposure. Since higher volatility means higher risk premiums, a wider variety of available financial products mitigates this risk.

Moreover, national budgets in the regions of oil-exporting are highly stressed, since global oil prices crashed because of global demand disruption. This means that oil exporters will desperately try to push up oil prices in a long-term view, since the current market share war is not sustainable for none of the market participants. An increase in oil prices would mean an increase in inflation. Therefore, a financial instrument which directly addresses and tracks the exposure of a CPI basket constituent, i.e. fuel, would appear to be a direct hedge for consumers against inflation.

All these factors effect negatively the business cycle, since there are just limited possibilities and instruments in the market available to hedge against increased fuel prices. Big companies, such as airlines, use standard market instruments for instance

crude oil futures in order to hedge their risk exposure. But other industries such as logistics, the trucking industry or private traffic sector are exempted from these possibilities, simply because of a missing direct exposure to fuel and minimum trade requirements for financial contracts.

We work in a filtered probability space $(\Omega, \mathscr{F}, \mathbb{P}, \mathscr{F}_{0\leq t<\infty})$, where the fuel price paths are assumed to be $(\mathscr{F}_t)$-adapted. The price dynamics can be described by the fuel price change dS_t. We further assume that the fuel price process under a real-world probability measure $\mathbb{P}$ takes the form

$$\frac{dS_t}{S_t} = \mu dt + \sigma(t, S)dW_t, \tag{32}$$

where the processes $\mu(t)_{0\leq t<\infty}$ and $\sigma(t)_{0\leq t<\infty}$ are $\mathscr{F}_t$-adapted, measurable and uniformly bounded. Furthermore, we assume that the fuel price satisfies the no-arbitrage condition. We deduct that all spot prices are observable since all future prices are observable as well. This implies absence of arbitrage, which in turn implies

$$\mathbb{E}_t^{\tilde{\mathbb{P}}}[dS_t] = (r_t^N - F_t)S(t)dt, \tag{33}$$

where r_t^N is the nominal risk-free rate and F_t is the instantaneous convenience yield. Therefore, the fuel price is assumed to follow

$$S_t = S_0 e^{Y_t}, \tag{34}$$

where (Y_t) is a $\mathbb{P}$-Brownian motion with drift. This means that for each t in $[0, \inf)$ we have

$$Y_t \sim N(\mu t, \sigma^2 t). \tag{35}$$

We further assume that S_t to be the value of the fuel price at time t, while Z_t denotes the value of an American call option at time t. What we are interested in, is the relation between these two prices at various exercise times.

Our methodology builds upon simplification of the application. Therefore, we operate in a discrete time frame, which proves to be more efficient. For this purpose we construct a binomial tree with T time steps corresponding to times $k= 0, 1, ...,T$, which reflects the fuel price S_k. Moreover, time T defines the maturity until a car driver or consumer intends to hedge his risk exposure, i.e. his fuel consumption. Since we are pricing an American option, the car driver has the right to exercise his fuel insurance at any given time prior to maturity. In our case, on a daily basis.

Our approach is based on the algorithm proposed by Mark Davis for calculating American options as in [17]. Since American options can be executed at any time, it is not possible to develop a closed formula for the pricing of such options. On each day the fuel price either increases to $S_{k+1}=uS_k$ or it decreases to $S_{k+1}=dS_k$ with $u> 1$ and $d=\frac{1}{u}$. Each increase or decrease is assumed to have the same probability

of occurrence. We let $\mathcal{F}_k$ be the σ-field generated by $\{S_0, S_1, ..., S_k\}$. At time k the possible price values are specified by a vector s_k=$s_k[0], ...,s_k[k]$ with

$$s_k[j] = u^k d^{2j} = d^{2j-k}. \tag{36}$$

Saved fuel is worth e^r at time T, given a risk-neutral environment. The probabilities for increases or decreases, p and d, are risk-neutral if the discounted price $\frac{S_k}{r^k}$ is a martingale. Thus, at time 0 this requires that S_0=$\frac{1}{e^r}(pu+(1-p)d)$ =$\mathbb{E}[\frac{S_1}{r}]$. Therefore,

$$p=\frac{e^r-d}{u-d}, \tag{37}$$

$$d=\frac{e^r-u}{d-u}. \tag{38}$$

Since we are pricing an American call option, the buyer has the right to buy fuel with exercise value $[S_k-K]^+$. Our pricing algorithm goes backwards and considers every possible exercise day by discounting every payoff at the end of each day.

In this paper, we propose a hedging approach for fuel. Since fuel prices are an important driver of inflation, we show that by valuing a U.S. call option, it is possible to hedge fuel for expected consumption within a given period in exchange for paying an appropriate premium.

In a risk-neutral environment, this premium depends on the inflation drift, which indicates an increase in commodity prices in general. This, in turn, strongly influences the determination of the price of an option, which is a derivative of an underlying, such as crude oil. We have shown two different frameworks in which such options could be priced, the continuous framework and the discrete framework.

For practical implementation, the discrete framework seems to be sufficient when considering daily time steps, since it is assumed that consumption is not necessary twice a day. Therefore, the exercise times can be reduced to daily time steps. This approach improves the computation time and provides a guideline.

Our results show that a consumer has to pay a premium of 6.7-8.4 cents per liter to hedge fuel prices for one year.

References

[1] ETF Specific Data Point Methodologies, Morningstar,*Morningstar Methodology Paper*, December 31, 2010.

[2] Secondary hedging for contingent claims, Stephan Unger and Kujtim Avdiu Journal of Financial and Economic Practice, Volume 13 Issue 2 Fall 2013, Bradley University

[3] Ingber, A. L. (1995), Adaptive simulated annealing (asa): Lessons learned, Control and Cybernetics.

[4] Henderson Brian J. and Pearson Neil D., 2011, The dark side of financial innovation: A case study of the pricing of a retail financial product, Journal of Financial Economics, Volume 100, Issue 2, pp. 227-247.

[5] Mikhailov, S. and Nögel, U. (2003), Hestons stochastic volatility model implementation, calibration and some extensions, Wilmott.

[6] Gatheral, J. (2004), Lecture 1: Stochastic volatility and local volatility, Case Studies in Financial Modelling Notes, Courant Institute of Mathematical Sciences.

[7] Heston, S. L. (1993), A closed-form solution for options with stochastic volatility with applications to bonds and currency options, The Review of Financial Studies 6 (2), 327-343.

[8] Stephan Unger and Lane P. Hughston, Stochastic Liquidity with Conditional Volatility, *Working paper*, Vienna University of Economics and Business, Imperial College London, 2010.

[9] Stephan Unger and Lane P. Hughston, Market Liquidity Measurement, *Working paper*, Vienna University of Economics and Business, Imperial College London, 2010.

[10] Stoimenov Pavel A. and Wilkens Sascha, 2005, Are structured products 'fairly' priced? An analysis of the German market for equity-linked instruments, Journal of Banking & Finance, Volume 29, Issue 12, December 2005, pp. 2971-2993.

[11] Bergstresser D., 2008, The retail market for structured notes: issuance patterns and performance, 1995â2008. Unpublished working paper, Harvard Business School.

[12] Kirkpatrick, S., Jr., C. D. G. and Vecchi, M. P. (1983), Optimization by simulated annealing, Science 220,(4598), 671-680.

[13] Brian C. Blanchett and David M. Blanchett, Tracking Tracking Error, *Journal of Indexes*, June 18, 2007, http://www.etf.com/publications/journalofindexes/joi-articles/2870.html.

[14] Roland Jeurissen and Jan van den Berg, Index Tracking using a Hybrid Genetic Algorithm, *Computational Intelligence Methods and Applications*, 2005 ICSC Congress, 10.1109/CIMA.2005.1662364.

[15] Rogalski R. and Seward J., 1991, Corporate issues of foreign currency exchange warrants Journal of Financial Economics, 30, pp. 347-366.

[16] Jarrow R. and O'Hara M., 1989, Primes and scores: an essay on market imperfections, Journal of Finance, 44, pp. 1263-1287.

[17] Davis H.A. Mark, American options in the binomial model (Revised), Finite Difference Methods, 2011.

2 Market Liquidity estimation in a high-frequency setup

Kujtim Avdiu [1]

[1] OeNB, Otto-Wagner-Platz 3, 1090 Vienna, Austria e-mail: kujtim.avdiu@oenb.at

September, 2017

Abstract

This article deals with the identification of a superior forecasting method for market liquidity using a calibrated Heston model for the bid/ask price path simulation instead of standard Brownian motion, and compound Poisson process, resp. inverse transform sampling for the generation of the bid/ask volume distribution. We show that the simulated trading volumes converge to one single value, which can be used as liquidity estimator and find that the calibrated Heston model as well as the inverse transform sampling are superior to the use of standard Brownian motion, resp. compound Poisson process.

Keywords: Market Liquidity, Heston model, Geometric Brownian motion, calibration, optimization techniques, Compound Poisson process,Market Liquidity,Inverse transformation sampling

2.1 Introduction

The paper claims to use optimization techniques in the context of financial models such as the Heston model. The goal is to apply these techniques to market liquidity modeling and identify any superior forecasting quality to the liquidity estimation approach using standard Brownian motion. The outstanding new idea within this work will be the application of various optimization techniques to the Heston model in order to analyze its forecasting quality in contrast to the standard Brownian motion. Its application to High frequency trading will add value to the novel approach of estimating market liquidity with Heston model. Financial models are characterized by a set of parameters which describe price movements of financial assets. In order to price and hedge financial products stoachstic models are most often used in practice and applied for general risk estimation frameworks.

Heston [16] developed a model for describing a movement of a stock price based on the standard Brownian motion approach but added an important component: stochastic volatility. Since a constant volatility is assumed in the classic framework it may not be appropriate to use such models in order to forecast future price developments. Heston's stochastic volatility model seems to be a better choice when modeling price fluctuations even though it makes assumptions which migh not be encountered in reality such as normal distributed returns in short time frames. The application of stochastic volatility models is very broad. In my thesis I will especially investigate the application of the Heston model to market liquidity simulation.

Market liquidity describes the ability of a financial market to absorb additional trades between market participants without affecting the price. When price impact is strong we speak of illiquidity of a market. In order to simulate market liquidity a certain parameter set is necessary to map trade capacity. This includes quoted bid and ask prices as well as quoted bid and ask volume as well as traded price and traded volume.

In order to simulate the Heston model as accurately as possible, a frequent update of the parameters used for estimation is necessary.

The Heston model has been used in different areas of the finance. These include pricing of derivatives and structured products, hedging, performance and risk estimation. Many articles address the application of the Heston model. Most works cover the computational implementation such as Ingber [12] or Kirkpatrick [13]. Others give summeries or overviews of the theoretical optimization techniques such as Mikhailov and Nögel[14] or Gatheral [15]. But no existing research has been done so far in order to apply a the Heston model to market liquidity measurement. Relying on market option data, fair values are calculated with the Heston model. Exact fair prices can only be achieved by making use of algorithmic optimization techniques. This optimization leads to a set of parameters which are then used to

simulate bid and ask prices of the Euro Stoxx 50 Future (FESX).
The bid and ask price simulations generated by the Heston model are applied to estimated traded volume in the FESX in order to test for accurateness. Due to the fact that a high percentage of trades are conducted at market and not in the limit order book, it is not possible to estimate expected traded liquidity just on the basis of historical quotations. In order to get an accurate estimation of traded volume it is necessary to scale the traded volume by the factor of 2. It can be shown that this factor turns out to be a characteristic number of the Compound Poisson process since just the minimum of a quoted bid and ask volume is traded. This characteristic number is subject to the estimation of traded volume quotations. This procedure of scaling is conducted in chapter 4 of my thesis.

The estimated traded volume serves as a good estimator for future expected market liquidity. Market liquidity is highly linked to market risk since illiquidity leads to high risk. Therefore it is obvious to link the risk classification number to the value of future expected market liquidity. Chapter 5 deals with the theoretic idea of making market liquidity tradeable. In that sense, the displayed value of future expected market liquidity could define the terms of trade between two counterparts who are willing to offset each other's opinion on expected traded volume. Since this value could be traded like a CFD this chapter focuses on the hedging of such positions if there is no counterpart to offset a trader's position.

Traded volume is an important factor for determining market liquidity. Since most liquidity is being generated at market and not via limit order book, estimation of at market volume is very difficult. The problem when simulating at market traded volume is that pointwise simulation leads to a non-homogenous movement which does not correspond to reality.

My research will focus on the computational implementation of a valid market liquidity estimation. The aim of my research will be to show that various simulation techniques exist for estimating future liquidity. It is likely that simulated traded volumes will not converge to the number of realized traded volumes due to the fact that the simulated quotes are Compound Poisson distributed random variables. Therefore an approximation algorithm might be necessary which generates values that are converging due to the law of large numbers.

The basis of my research within this chapter will be the current working paper of Unger and Hughston[19] which investigates the liquidity generating price process. It incorporates two standard Brownian motions which represent the price process of the buyer and the seller. The intersection of the two Brownian motions serves as the indicator for the corresponding volume generating process and therefore as the indicator variable for the points in time when liquidity is being generated. A simulated trade occurs by taking the minimum of two quoted sizes. This means if a buyer wants to buy say 50 units to a certain price and a seller agrees on that price,

but only wants to trade 10 units, liquidity of 10 units is being generated. Since the quoted volume is assumed to be simulated by a Compound Poisson process, the law of large numbers yields exponentially distributed random numbers. The interesting question will be how the convergence process according to the law of large numbers looks like if we take the minimum of two exponentially distributed random variables.

The usual way to describe the price dynamics of an asset price process is to assume conditions such as stochastic independence of increments, finite variation and driving factors. Such factors may encompass a drift term, stochastic volatility, speed of mean reversion or correlation terms. In order to generate a price process which is close to reality it is necessary to update the randomized price movement with the implied volatilities and its subsequent traded prices of the corresponding derivatives. This means, without having a method to reduce the error between the estimated value and the realized value, it is impossible to stick to the real price development.

The problem with stochastic models is that there doesn't exist a closed-form solution for every kind of model. In such a case numerical solutions are needed. These numerical computations need to be calibrated to current market data. Since regular calibration techniques need a lot of computational resources the focus of this work lies in the application of robust calibration.

One important calibration parameter which is widely used for pricing financial products is implied volatility. The Heston model assumes stochastic volatility which seems to reflect reality better than constant volatility, as used in the classic Black-Scholes framework. The standard approach is the least-square type calibration. The shortfall of this approach is its sensitivity to the choice of the initial point: The point of convergence depends on the point of departure.

The goal of this chapter will be to calculate the fair value of Euro Stoxx 50 Future options based on the Heston model. On basis of the obtained parameters a simulation of expected traded volume will be performed. The parameters needed for estimation of the volatility parameter v_0 are $\{\kappa, \theta, \sigma, \rho\}$, where κ is the mean reversion rate, θ denotes the long run variance, σ the volatility and ρ the correlation. The error resulting from the least square calibration is subject to optimization procedures.

For reducing this error several optimization methods exist, such as Genetic algorithms (GA), the Particle Swarm Optimization technique (PSO), the Levenberg-Marquardt method (LM) and the Nelder-Mead Simplex method (SM). Not many papers have dealt with these kind of optimization techniques in terms of financial market data calibration. The application to liquidity estimation outlines the novelty of our research in this context.

Genetic algorithms are e.g. used by Poklewski-Koziell [23] for calibrating the Heston model to synthetically generated data. GAs are based on the natural selection

and evolution, where the stronger individuals of a population are selected over the weaker ones. By making us e of this concept, the GA optimizes the relevant parameters for the Heston model by evaluating how well the individual parameters in the parameter space optimize the objective function. These individual parameters are assigned fitness values based on how well the difference to the objective function is minimized. Well fitting parameters are allowed to reproduce in order to create subsequent population generations.

Particle Swarm Optimization technique is a self-learning algorithm for localizing minima in multidimensional spaces that outperforms traditional sampling methods in terms of computational cost. [36] Gilli and Schumann [22] calibrate option prices by application of PSO to the Heston model. Kuok and Chan [21] have applied PSO for calibration in a conceptual hydro-logical context. Other field of application cover areas such as cosmology and extra-galactic astrophysics. Ruiz et al.[36] apply PSO to comparing a new method called semi-analytic models of galaxy formation and evolution (SAMs) in order to test the consequences of including new astrophysical processes in galaxy formation models.

The Levenberg-Marquardt method is the industry standard when it comes to optimization of multivariate non-linear systems posed as least squared problems. It approximates the non-linear system near the minimum with a quadratic system.[24] LM is an iterative technique that locates the minimum of a function that is expressed as the sum of squares of non-linear functions [25]. In our context its application is to optimize the relevant Heston parameters so that the sum of squares of the deviations to the realized price values becomes minimal.

The Nelder-Mead Simplex method is based on a concept of a geometric object, a so called simplex. In 2D this is a triangle, in 3D it is a pyramid. The idea behind it is a non-linear optimization method which allows to generate a new point in each iteration in or near the geometric object. To determine the location of this new point, a reflection step is introduced where the new point is chosen to be the reflection of the worst existing point. By doing so, the simplex is moving away from the high energy landscape to the low energy landscape. The algorithm stops if the simplex is small enough. Then the solution is a point inside of the simplex.[32]

All these optimization techniques are performed in order to minimize the estimation error between the Heston option price and the realized option price under the condition of minimizing the sum of the squared percentage errors between model and market implied volatilities. [27][28]

For the estimation of the liquidity, the simulation of bid and ask volumes is necessary. For this purpose, we present two methods, Compound Poisson process and Inverse transform sampling. We examine these two methods with a Chi-Square test.

2.2 Model description

The starting point of our research is the liquidity intersection model by Unger and Hughston.[26] It defines a liquidity generating process based on 4 key parameters prevailing on market: Bid price, ask price, bid volume and ask volume. The mechanics is is follows: The bid and ask prices are simulated independently by geometric Brownian motions as well as the corresponding bid and ask volume processes which are simulated by compound Poisson processes. Every time the bid price reaches the level of the ask price or exceeds it, a stopping time τ is defined. This stopping time refers to the time where liquidity is generated by the minimum of the prevailing bid and ask quoted volumes. By repitition and averaging of the procedure, the simulated traded volume converges to one single value, depending on the bid/ask spread. This singular value is the estimated liquidity for the time period of simulation of the corresponding asset.

The described simulation is conducted under assumption of normal distributed random varibles and by estimation of μ and σ. These are the input parameters for the geometric Brownian motions which serve as the driving random price processes of the bid and the ask price. The procedure applies the Black-Scholes framework and all its properties. The point of interest is where we assume a different underlying price process of the liquidity generating process. For this purpose we apply the Heston model and show how the liquidity value behaves when the Heston model is calibrated with four different optimization methods. These are the already mentioned Genetic Algorithms (GA), the Particle Swarm optimization technique (PSO), the Levenberg-Marquardt method (LM) and the Simplex method (SM).

For the estimation of market liquidity we assume two stochastic processes for the development of the bid (i) and ask (j) prices, two compound stochastic processes for the simulation of bid (i) and ask (j) volumes and an algorithm for the calculation of the arithmetic mean of the traded volumes over a certain period of time. The arithmetic mean is calculated by taking the generated traded volume in a certain time interval, divided by the total number of time steps.

2.2.1 Simulation of bid (i) and ask (j) prices

For the simulation of the Prices we propose Geometric Brownian motion [20] and Heston model and compare the estimated results.

The multivariate geometric Brownian motions for the simulation of the price process of the buyer S_j and the price process of the seller S_j with $\mu_{i/j}$(bid/ask price drift) and $\sigma_{i/j}$(price volatility) can be developed as follows:

$$S_i(t) = S_i(0)\, e^{\mu_i t + \sigma_i X_i} \tag{39}$$

$$S_j(t) = S_j(0)\, e^{\mu_j t + \sigma_j X_j} \tag{40}$$

where the correlation matrix between the Wiener processes X_i and X_j is defined as $E(X_i, X_j) = \rho_{i,j}$ and $\rho_{i,i} = 1$.

The Heston model assumes a stochastic volatility development of the bid (i) and ask (j) prices ($S_{i,j}$) with parameters $\mu_{i,j}$(bid/ask price drift), $V(t)_{i,j}$(bid/ask price variance), κ (rate of mean reversion), $\omega_{i,j}$(long run variance), $\sigma_{i,j}$(volatility of variance) and $W_{1,2}^{i,j}$ (Standard Brownian movements). Thus we take for the bid dynamics:

$$\frac{dS(t)_i}{S(t)_i} = \mu_i dt + \sqrt{V(t)_i} dW_1^i, \tag{41}$$

$$dV_i = \kappa_i(\omega_i - \sigma_i)dt + \sigma(t)_i \sqrt{V(t)_i} dW_2^i \tag{42}$$

and for the ask price dynamics:

$$\frac{dS(t)_j}{S(t)_j} = \mu_j dt + \sqrt{V_j} dW_1^j, \tag{43}$$

$$dV_j = \kappa_j(\omega_j - \sigma_j)dt + \sigma(t)_j \sqrt{V(t)_j} dW_2^j, \tag{44}$$

where W_1 and W_2 are correlated by $dW_1 \cdot dW_2 = \rho dt$ due to the leverage effect between asset price and instantaneous volatility.

2.2.2 Simulation of bid (i) and ask (j) volumes

Parallel to the bid and ask price process we have two corresponding bid and ask volume processes running. Therefore we propose 2 methods:

Compound Poisson process

The dynamics of the volumes processes Q_i and Q_j are assumed to follow a compound Poisson process with jump rate λ. Thus for the bid and ask volume processes we have

$$P(Q_i(t) = n) = e^{-\lambda_i t} \frac{(\lambda_i t)^n}{n!}, \tag{45}$$

$$P(Q_j(t) = n) = e^{-\lambda_j t} \frac{(\lambda_j t)^n}{n!}, \tag{46}$$

where Q_i resp. Q_j denotes the submitted volume quote a buyer or a seller wants to buy resp. sell at time t.

Inverse transform sampling

The dynamics of the volumes processes Q_i and Q_j are assumed to be simulated with a inverse transform sampling as follows:

Let

$$P(X_i = x_i^k) = p_i^k, k = 1, 2, ...n_i, with \sum_{k=1}^{n_i} p_i^k = 1 \tag{47}$$

$$P(X_j = x_j^k) = p_j^k, k = 1, 2, ...n_j, with \sum_{k=1}^{n_j} p_j^k = 1 \tag{48}$$

be the probability mass function (PMF) of of bid (i) resp. ask (j) volume calculated by historical data with n_i quoted bid volumes $x_i^k, k = 1...n_i$ and n_j quoted ask volumes $x_j^k, k = 1...n_j$. Then the PMF of quoted bid (i) resp. ask (j) volume at time t is given by

$$P(Q_i(t) = x_i^k) = P(F(x_i^{k-1}) \leq U \leq F(x_i^k)) \tag{49}$$

$$P(Q_j(t) = x_j^k) = P(F(x_j^{k-1}) \leq U \leq F(x_j^k)) \tag{50}$$

with $F(x_{i/j}^t)$ the cumulative distribution function (CDF) of historical quoted bid (i) and quoted ask (j) volumes at time t calculated from PMF

$$F(x_{i/j}^k) = \sum_{t=1}^{k} p_{i/j}^t \tag{51}$$

and U an uniform random variable between 0 and 1.

Since for $x, y \in [0, 1]$ and $x \leq y$

$$P(x \leq X \leq y) = y - x \tag{52}$$

we can easly prove that the Inverse transform sampling could be a good estimation for the development of bid (i) and ask (j) volume processes

$$P(Q_{i/j}(t) = x_{i/j}^k) = \sum_{t=1}^{k} p_{i/j}^t - \sum_{t=1}^{k-1} p_{i/j}^t = p_{i/j}^k \tag{53}$$

2.2.3 Liquidity Estimation

The resulting compound volume process $Y(t)$ at time t characterizes the volume generating process induced by trading.

$$Y(t) = \begin{cases} min\{Q_i(t), Q_j(t)\} & \text{if } S_i(t) \geq S_j(t) \\ 0 & o.w. \end{cases} . \tag{54}$$

The result is a non-homogeneous compound process. The *min* -condition ensures that the minimum amount of two matched quantities is listed as a transaction, or traded volume, responsible for the market impact and its costs. By matching bid and ask prices and taking the average of each possible generated volume we get the average traded volume (liquidity) over a certain time period n:

$$E[Y] = \frac{\sum_{t=1}^{n} Y(t)}{n} \tag{55}$$

2.3 Optimization

For simulation of the bid and ask prices we proposed geometric Brownian motion and Heston model.

Since BGM has the lognormal distribution with parameters $(\mu - \frac{\sigma^2}{2}t)$ and $\sigma\sqrt{t}$ for $t \in [0, \infty]$ we can use the average mean value as estimator for μ and standard deviation as estimator for σ, calculated from the historical data.

Heston model is based on the assumption that the volatility of the underlying is stochastic and includes more parameters to be estimated. To conduct an estimation we use the theoretical plain vanilla bid/ask option price. We proceed by developing an analytic expression for the Fourier transform of the option price and then get the price back by Fourier inversion.

For optimizing the order submission flow on the ask-volume as well as on the bid-volume side we perform a Chi-Square optimization. Our null-hypothesis states that the real arrival times of conducted trades follow a Compound Poisson distribution or an inverse Fourier transformation.

2.3.1 Optimization of Heston parameters

In order to validate the effectiveness of the Heston model optimization we estimate the Heston parameters by using four different techniques: 1. Genetic algorithms, 2. Particle Swarm Optimization 3. Levenberg-Marquardt and 4. Nelder-Mead Simplex.

We perform these optimization techniques in order to minimize the root mean square error between the estimated plain vanilla Heston option price and the realized option price of option i on the estimation day:

$$arg\ min_{\Omega} \sqrt{\frac{\sum_{i=1}^{N}(C_0(i, r, M_i, S, K_i) - C_i)^2}{N}}, \tag{56}$$

where Ω is the set of heston parameters to be estimated, N the number of options on the estimation day, C_0 is the heston call function denotes the dollar adjusted call

plain vanilla option price, r interest rate, M_i the Maturity of Option i, S the closing price of the Underlying, K_i the strike of option i.

In order to calculate the plain vanilla option prices e.g. Fast Fourier Transform (FFT) can be applied.[29] Assuming no dividends and constant interest rate r, the initial option value C_0 is

$$C_0 = S\Pi_1 - Ke^{-rT}\Pi_2, \tag{57}$$

with risk-neutral probability that the option matures in-the-money:

$$P(S_T > K) = \Pi_2 = \frac{1}{2}\frac{1}{\pi}\int_0^\infty Re(\frac{e^{iuln(K)}\phi_T(u)}{iu})du \tag{58}$$

and the delta of the option

$$\Pi_1 = \frac{1}{2}\frac{1}{\pi}\int_0^\infty Re(\frac{e^{iuln(K)}\phi_T(u-1)}{iu\phi_T(-i)})du, [29] \tag{59}$$

where $\phi_T(u)$ denotes the characteristic function of $ln(S_T)$, which is

$$\phi_T(u) = \int_{-\infty}^{infty} e^{ius}q_T(s)ds, \tag{60}$$

q_T denoting the risk-neutral probability of the log-price of the underlying.[30]

Since the integrand is singular at $u = 0$, FFT can't be used for evaluating the integral. But to make use of the speed advantage of FFT, we make use of the relation of the initial call option value $C_T(k)$ and the risk-neutral density $q_T(s)$ by

$$C_T(k) = \int_k^\infty e^{-rT}(e^s - e^k)q_T(s)ds. \tag{61}$$

$C_T(k)$ tends to S_0 as k tends to ∞ which indicates that the call price function is not square-integrable. But the FFT can be applied when using the modified call price $c_T(k)$ defined by

$$c_{T(k)} = exp(\alpha k)C_{T(k)} \text{ for } \alpha > 0. \tag{62}$$

With the invesere transform we can get $C_T(k)$ by

$$C_T(k) = \frac{exp(-\alpha k)}{\pi}\int_0^\infty e^{-ivk}\Phi(\nu)d\nu, \tag{63}$$

where

$$\Phi_T(\nu) = \frac{e^{-rT}\phi_T(\nu - (\alpha+1)i)}{\alpha^2 + \alpha - \nu^2 + i(2\alpha+1)\nu}. \tag{64}$$

For out-of-the money options, the call price can be obtained with the Fourier transform $\xi_T(\nu)$ of $z_T(k)$ for

$$\xi_T(\nu) = \int_{-\infty}^\infty e^{i\nu k}z_T(k)dk. \tag{65}$$

By inversion of this transform we get

$$z_T(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\nu k} \xi_T(\nu) d\nu, \tag{66}$$

with

$$\xi_T(\nu) = e^{(-rT)}\left(\frac{1}{i+i\nu} - \frac{e^{rT}}{i\nu} - \frac{\phi_T(\nu-i)}{\nu^2 - i\nu}\right).[29] \tag{67}$$

Since we are now able to generate option prices based on stochastic volatility we can now simulate the bid and ask of the stock price using Heston model.

The calibration of the Heston parameters follows the approach of the multi asset Heston model. This allows us to simulate the development of bid and ask prices taking into account the high correlation. [37]

Since the quoted bid and ask prices in the order book do not always correspond to realized traded prices (too high or too low) a selection of the optimal options for the calibration of the Heston model is necessary. For this reason, we have tested options by maturity and moneyness levels as well as the optimal amount of options in the sample and the historic sampling period.

In order to solve the optimization problem we apply four different techniques and compare their accuracy within the Heston model to the realized historical prices of the options.

2.3.2 Heston model calibration with Genetic algorithms

Genetic algorithms operate by maintaining and modifying the characteristics of a set of trial solutions (individuals) over a number of iterations (or generations). Each individual solution is represented by a binary string (or chromosome in biological analogies) in the genetic algorithm. The optimization process is designed to produce, in successive iterations, an increasing number of individuals with desirable characteristics. The process is probabilistic but not completely random. The rules of genetic algorithms have the power of retaining certain desirable characteristics that would otherwise be lost with a completely random searching method. A genetic algorithm consists of the following major components: encoding method, reproduction, crossover, mutation and fitness scale.[9]

Encoding

The initial step in operating the genetic algorithms is to form an initial trial solution set. Each Heston model parameter is encoded in the form of a binary sub-string consisting of '0' and '1'. The length of a sub-string depends on the accuracy required in the model parameter that it represents. The sub-strings are linked together to form one binary string. If one start with 50 trial solutions, then there would be

50 binary strings. Each string represents a set of Heston model parameters that, if decoded, could be used to calculate the value of the objective function. In the initial set, the values of the Heston model parameter are randomly assigned.[9]

Reproduction

The initial set of strings would generally not provide an optimal solution. Genetic algorithm works by trying to generate (reproduce) other strings that would fit better the objective function. The reproduction process is simply a selection process of those strings having a better objective function will have a higher chance of reproduction.[9]

Crossover

The crossover process in which genetic materials (the binary bits) between strings are exchanged is used to generate new strings. Various crossover methods such as single-point crossover, multiple-point crossover and uniform crossover could be used.[9]

Mutation

To avoid being trapped into a local optimal point, a mutation process is used. In this process, some bits in the strings are selected randomly and their values are changed. Mutation is generally damaging rather than beneficial to the optimization process. However, it provides a mechanism for the search to jump out of a local optimum point.[9]

Fitness scale

During the implementation of the genetic algorithm, a particular type of string may be excessively reproduced and the algorithm would then converge prematurely, generating unreliable results. After a number of iterations, the strings in the set tend to become very similar and hence the variation of objective values associated with the strings is very small. The relative importance of these strings cannot be distinguished during the reproduction process. Therefore the desirable (outstanding) string within the trial solution set cannot be easily identified. in order to overcome this problem and to increase the variation in objective values, a fitness scale is applied to linearly map the objective values to a pre-specified range so as to enhance the resolution for desirable strings.[31]

2.3.3 Heston model calibration with Particle Swarm Optimization

In Particle Swarm Optimization (ps; Eberhart and Kennedy, 1995), we have again a population that comprises n solutions, stored in real-valued vectors. In every generation, a solution is updated by adding another vector called velocity v_i. We may think of a solution as a position in the search space, and of velocity as a direction into which the solution is moved. Velocity changes over the course of the optimization, the magnitude of change is the sum of two components: The direction towards the best solution found so far by the particular solution, P_{best_i} , and the direction towards the best solution of the whole population, $P_{best_{gbest}}$. These two directions are perturbed via multiplication with a uniform random variable ξ and constants $c_{(\ddot{\imath}^{\shortparallel}\cdot)}$, and summed. The vector obtained is added to the previous v_i, the resulting updated velocity is added to the respective solution. In some implementations, the velocities are reduced in every generation by setting the parameter Δ, called inertia, to a value smaller than unity. According to Gilli and Schumann [22] the algorithm for the differential evolution can be summarized as follows:

Algorithm

1. set parameters n_P, n_G, F and CR
2. Initialize population $P^{(1)}_{j,i}$, $j = 1, ..., p$, $i = 1, ..., n_P$
3. **if** $k = 1$ to n_G **do**
4. $\quad P^{(0)} = P^{(1)}$
5. $\quad$ **for** $i = 1$ to n_P **do**
6. $\quad\quad$ **generate** $l_1, l_2, l_3 \in \{1, ..., n_P\}$, $l_1 \neq l_2 \neq l_3 \neq i$
7. $\quad\quad$ **compute** $P^{(v)}_{.,i} = P^{(0)}_{.,l_1} + F \times (P^{(0)}_{.,l_2} - P^{(0)}_{.,l_3})$
8. $\quad\quad$ **for** $j = 1$ to p **do**
9. $\quad\quad\quad$ **if** $\xi < CR$ **then** $P^{(u)}_{j,i} = P^{(v)}_{j,i}$ **else** $P^{(u)}_{j,i} = P^{(0)}_{j,i}$
10. $\quad\quad$ **end for**
11. $\quad\quad$ **if** $F(P^{(u)}_{.,i}) < F(P^{(0)}_{.,i})$ **then** $P^{(1)}_{.,i} = P^{(u)}_{.,i}$ **else** $P^{(1)}_{.,i} = P^{(0)}_{.,i}$
12. $\quad$ **end for**
13. **end for**
14. find best solution $gbest = argmin_i\ F(P^{(1)}_{.,i})$
15. $solution = P^{(1)}_{.,gbest}$

where n_G is the number of generations, P is the population (a matrix of size $p \times n_P$), F is the objective function, F_i is the objective function value associated with the ith solution and xi is a random variate with uniform distribution on $[0, 1]$.

2.3.4 Heston model calibration with Levenberg-Marquardt

LM is an iterative technique that locates the minimum of a function that is expressed as the sum of squares of nonlinear functions. Its principal application is to optimize the parameters $f(x) = \{\kappa, \theta, \sigma, \rho\}$ of the Heston model curve $f(y, x)$, where y is a given variable, so that the sum of the squares of the deviations $S(x)$ becomes minimal.

The Steepest Gradient Descent Method

The steepest descent method is a general minimization method which updates parameter values in the direction opposite to the gradient of the objective function. [34] The steepest gradient descent method works by making a step t that is the negative gradient of the error times some constant.

$$\Delta B = -a\frac{\partial E(B)}{\partial B}, \tag{68}$$

where B_0 defines the initial guess of the parameter set and $E = T - f(z_k, B_j)$ is the estimation error function with

- T is the experimental data,
- $f(z_k, B_j)$ is the fitting function,
- z_k are the number of independent data of dimension N,
- B_j are the parameters of dimension M,
- a is a positive constant.

This means that in steep regions (where slow convergence is advisable) the algorithm moves quickly and in shallow regions (where fast convergence is more favorable) the method moves slowly. The steepest gradient descent method works fine with simple models, but it fails when more complexity is added. In addition, convergence can take a long time because the method goes through most of the error surface missing the minima. [33]

The Gauss-Newton Method

The Gauss-Newton method is a method for minimizing a sum-of-squares objective function. It presumes that the objective function is approximately quadratic in the parameters near the optimal solution. [34] The method is based on the idea that nonlinear models can be approximated by linear functions through Taylor expansion when the system is close to a minimum in error space. Then the square error (E^2) will approximate a quadratic equation where the linear least square method can be used to find a minimum.

Levenberg-Marquardt

The Levenberg-Marquardt method performs an interpolation between the Gauss and the steepest gradient descent methods based upon the maximum neighborhood in which the truncated Taylor series gives an adequate representation of the nonlinear model. [35]

In the algorithm a positive constant (damping) is added to the diagonal of of A^TA in order to control the convergence of the method and provide an effective way to avoid the singularity of the system. In the former case damping will determine the rapidness of convergence, with large damping producing slow convergence and vice versa. In the latter case the presence of damping will artificially increase the eigenvalues improving the ill-conditioning of matrix A^TA. In the method the step to converge from an initial guess to a final solution is represented by

$$\Delta B = (A^TA + \epsilon^2 I)^{-1} A^T \Delta T, \tag{69}$$

where

- B= Parameters to find,
- A= Jacobian matrix,
- ϵ^2=Damping,
- T=Data.

The general steps of the Levenberg-Marquardt algorithm are as follows:

Algorithm

1. Choose the initial parameters B_0
2. Choose the values for the positive constants α and β
3. Start with a large initial damping ϵ_0^2
4. Determine A^TA
5. Determine ΔB and calculate B_{i+1}
6. Check at each step
   -if $RMSE_i < RMSE_{i-1}$, then $\epsilon_{i+1}^2 = \epsilon_i^2/\beta$
   -if $RMSE_i > RMSE_{i-1}$, then $\epsilon_{i+1}^2 = \alpha\epsilon_i^2$
7. Maintain a minimum value for damping (ϵ_{min}) to ensure non-singularity of the matrix $Q = A^TA + \epsilon^2 I$. [33]

2.3.5 Heston model calibration with Nelder-Mead Simplex

The Nelder-Mead Simplex method or Downhill Simplex method is a direct search method which is independent of the existence of derivatives. It solves the optimization problem given a real-valued objective function by starting from a simplex S_0. For each iteration step indexed by k the algorithm identifies a vertex v^k_{max} determined by

$$v^k_{max} = argmax_{x\in\{v^k_0,...v^k_n\}f(x)}, \tag{70}$$

which is the vertex where the function takes its largest value. The vertex v^k_{max} of the simplex S_k is then replaced by a new point $\hat{v}$ such that $f(\hat{v}) < f(v^k_{max})$. The new simplex is called S_{k+1}. To construct the new vertex $\hat{v}$ of this simplex the method uses three operations called reflection, expansion and contraction.[11] For our purpose we apply the Downhill Simplex method to the given parameter set of the Heston model.

Algorithm

Step 1. Initialize $0 < \alpha \leq 1$, $0 < \beta < 1$, $\gamma > 1$ and the ordered vertices v_j for $j = 0, ..., n$ of simplex S_0. Choose a maximum number of iterations k_{max} and set $k := 0$.

Step 2. **if** $k > k_{max}$ **then**
stop

Step 3. Calculate the centroid $\hat{v} := \frac{1}{n}\sum_{i=0}^{n-1} v_i$

Step 4. Reflection: Calculate $x_r = (1+\alpha)\hat{v} - \alpha v_n$ and set $v^k = x_r$.

if $f(x_r) < f(v_{n-1})$ **then**

Step5. **if** $f(x_r) < f(v_0)$ **then** Expansion:
Compute $x_e = (1-\gamma)v_n + \gamma x_r$
if $f(x_e) < f(x_r)$ **then**
set$v^k = x_e$

Step 6.**else** Contraction:
if $f(v_{n-1}) \leq f(x_r) < f(v_n)$ **then** Partial Outside:
Set $x_c = ()1-\beta)\hat{v} + \beta x_r$
else $f(x_r) \geq f(v_n)$ Partial Inside:
Set $x_c = ()1-\beta)\hat{v} + \beta v_n$
if $f(x_c) < f(v_n)$ **then**
Set $v^k = x_c$
else Total Contraction:
for $j = 1$ **to** n **do**
$v_j = \frac{v_0+v_j}{2}$
and set $v^k = v_n$

Step 7. Update simplex S_{k+1} by sorting $\{v_0, v_1, ..., v_n\}$. Set $k := k+1$ and go to Step 2.

In order to prevent the algorithm of getting trapped in an endless loop it is possible to pre-define a number of iterations maximum allowed k_{max} or to extend additional stopping criteria taking into account the change in the values of the objective function values or the size of the current simplex. Nelder and Mead propose using the standard deviation

$$\sigma_f := (\frac{1}{n+1}\sum_{i=1}^{n}(f(v_i) - \bar{f_k})^2)^{\frac{1}{2}} \tag{71}$$

with

$$\bar{f_k} = \frac{1}{n+1}\sum_{j=1}^{n} f(v_j) \tag{72}$$

of the objective function evaluated on the vertices of the simplex S_k as a stopping rule. If σ_f is below a sufficient small chosen tolerance level $\epsilon > 0$, the algorithm stops. The optimal solution is then given by vertex v_0. [11]

2.3.6 Heston model calibration with Levenberg-Marquardt combined with genetic algorithms

Due to the large amount of data, high frequency of the simulations, and complexity of the closed Heston formula for calculating option prices, the computer run times for optimizing Heston parameters are very high. For this reason, we have chosen weak termination conditions (Maximum function evaluation, maximum iterations, termination tolerance on the function value). The calibration results are very satisfactory for all optimization methods.

Nevertheless, we also tested a combination of Levenberg-Marquardt and Genetic Algorithm by using the optimal parameters of LM as starting parameters for the GA. This allows us to improve the optimization error (see results below).

2.3.7 Chi-Square optimization test of bid and ask volume

For the simulation of the order submission flow we test for two different arrival time distributions for the ask as well as for the bis side: Compound Poisson - distributed and inverse-transformed arrival times. Our null-hypothesis states that these simulated arrival-times correspond to the real order submission arrival times:

- H_0: The bid-ask volume generated by compound Poisson process resp. inverse transform has a statistically significant association with the cumulative distribution function of the real observed bid-ask volume data.
- H_A: The bid-ask volume generated by Compound Poisson process or inverse transform is distinct from the cumulative distribution function of the real observed bid-ask volume data.

H_0 is accepted if

$$1 - pvalue = P[\chi^2 > \chi^2_{0.05}|H_0] > 0.05 \tag{73}$$

In order to test this assumption we conduct a Chi-Square test for different time intervals and take the arrival times which matches best reality.

The Compound Poisson process is a distribution which consists of 2 parameters, λ and n, which measures the arrival time and volume. To estimate λ we take a sample length of real order submissions which serves as the mean arrival time for our simulation:

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{74}$$

where x_i are the order submissions per time unit. λ serves as a good estimator for the maximum likelihood method. To estimate we take historical data and calculate the mean value of bid/ask volumes for a specific time period.

For the inverse-transformed distribution we take random sample from the past based on the PMF.

In order to perform a Chi-Square test for simulated ask and bid order submission we classify for both methods:

- a_i^o: Number of ask order submissions,
- a_i^e: Expected number of ask order submissions

for the ask volume side as well as

- b_i^o: Number of bid order submissions,
- b_i^e: Expected number of bid order submissions

for the bid volume side and conduct the test

$$\chi^2_a = \frac{(a_i^o - a_i^e)^2}{a_i^e} \tag{75}$$

for the ask-volume side as well as

$$\chi^2_b = \frac{(b_i^o - b_i^e)^2}{b_i^e} \tag{76}$$

for the bid volume side.

2.4 Validation of parameter optimization

This section provides the results of the parameter estimation using FESX50 Future option data as well as the statistical test results from the bid-ask volume Chi-Square optimization. For the FESX50 parameter optimization we compare root mean square errors of the four different optimization techniques, namely Genetic algorithms (GA), Particle Swarm Optimization (PSO), Levenberg-Marquardt (LM), and Nelder-Mead Simplex (NM). The parameters of interest are the greeks, κ, θ, σ, ρ , and v_0 needed to determine the price of an option, priced with the Heston model.

For the bid-ask volume Chi-Square optimization we provide in-sample as well as out-of sample test results for Compound Poisson process and inverse transformation.

2.4.1 Validation of Heston optimization

Data overview

The time frame used for testing is 27.04.2015-30.12.2015, the source for the option data is Interactive Brokers, and the source for the Euribor interest rate data is Bloomberg. We use 1301 Call options with different maturities and different strikes. On average 70 to 80 options were traded daily.

Table 5: Data example for August 20, 2015

Maturity (years)	Strike Price	Stock Price	Last Bid	Last Ask
1.92	3,318	3,800	103.6	104.2
1.92	3,318	3,000	453.4	454.7
3.37	3,318	4,100	102.0	102.4
3.37	3,318	4,300	74.9	75.0
1.92	3,318	3,300	294.1	295.4
1.92	3,318	3,400	247.0	247.9
1.92	3,318	2,500	836.9	838.2
3.37	3,318	2,500	805.5	806.1
4.84	3,318	3,800	214.5	215.2
6.28	3,318	3,600	310.0	311.0
1.20	3,318	3,050	388.2	389.6
1.20	3,318	3,500	157.0	157.5
1.20	3,318	3,550	137.0	137.9
1.92	3,318	3,250	319.8	321.0
1.92	3,318	3,350	269.7	270.8
1.20	3,318	3,600	113.6	113.6

Continued on next page

Table 5 – *Continued from previous page*

Maturity (years)	Strike Price	Stock Price	Last Bid	Last Ask
1.92	3,318	3,650	151.0	151.7
1.20	3,318	3,750	74.5	74.6
1.20	3,318	3,800	58.0	58.3
1.92	3,318	3,850	98.5	98.9
1.20	3,318	3,900	43.0	43.1
1.20	3,318	4,000	26.6	26.8
2.64	3,318	3,500	215.0	215.2
2.64	3,318	4,000	84.0	84.2
1.20	3,318	4,100	19.3	19.4
1.20	3,318	4,400	6.1	6.1
1.20	3,318	4,500	3.2	3.2
1.20	3,318	4,600	1.6	1.6
1.20	3,318	4,700	1.8	1.8
1.20	3,318	5,000	0.9	0.9
1.20	3,318	4,150	19.6	19.6
4.84	3,318	4,300	133.0	133.3
0.84	3,318	3,300	237.4	238.6
0.84	3,318	3,400	203.0	204.2
0.84	3,318	3,450	179.4	180.7
0.84	3,318	3,500	158.5	159.1
0.84	3,318	3,550	136.5	137.4
0.84	3,318	3,600	114.0	114.1
0.84	3,318	3,650	88.7	89.2
0.84	3,318	3,700	77.9	78.3
0.84	3,318	3,800	49.0	49.1
0.84	3,318	3,850	42.9	43.1
0.84	3,318	3,900	34.4	34.4
0.84	3,318	3,950	26.7	26.8
0.84	3,318	4,000	20.0	20.1
0.84	3,318	4,100	12.0	12.0
0.84	3,318	4,200	7.4	7.4
0.84	3,318	4,250	6.4	6.4
0.84	3,318	4,300	5.1	5.1
0.84	3,318	4,350	3.8	3.8
0.84	3,318	4,400	2.8	2.8
0.84	3,318	4,450	2.4	2.4
0.84	3,318	4,600	1.5	1.5
0.84	3,318	4,500	2.0	2.0

Continued on next page

Table 5 – *Continued from previous page*

Maturity (years)	**Strike Price**	**Stock Price**	**Last Bid**	**Last Ask**
1.20	3,318	4,550	2.6	2.6

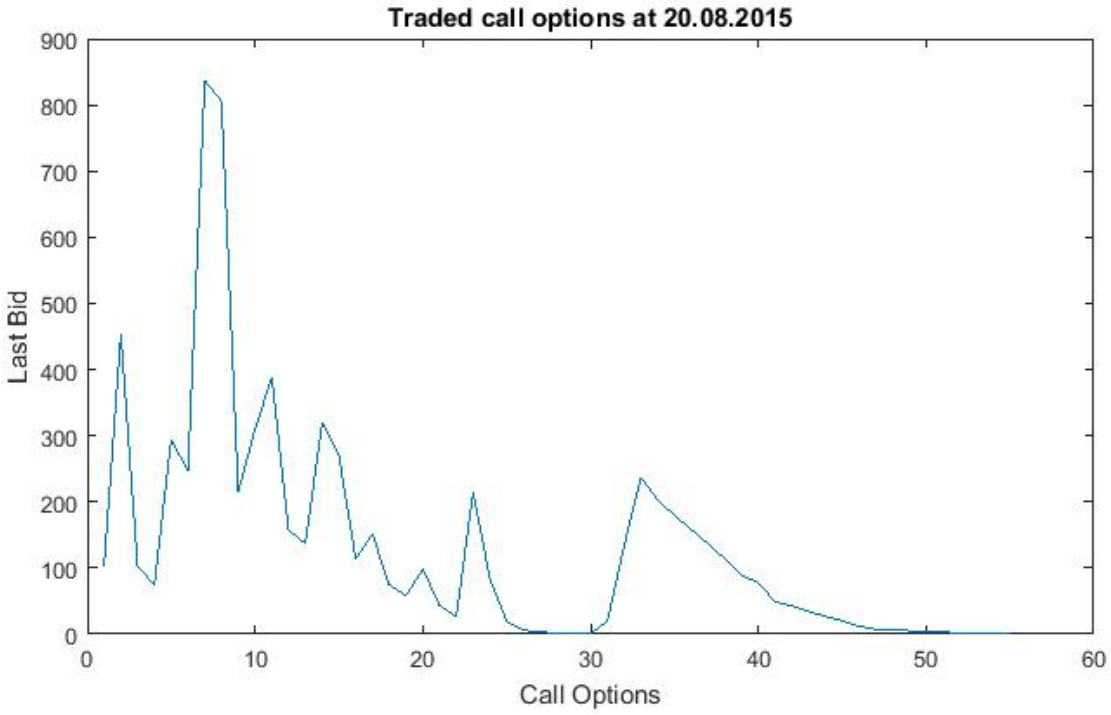

Figure 1: End of day last bid prices of all traded call option on August 20, 2015.

Table 6: Euribor interest rates on August 20, 2015

Maturity (years)	Euribor rate
0.02	-0.14%
0.05	-0.13%
0.08	-0.09%
0.17	-0.05%
0.25	-0.03%
0.50	0.04%
0.75	0.09%
1.00	0.16%

For the estimation of the parameters we need to use different maturities. For calculation we do a linear interpolation by

$$r(t) = r(t+1) - r(t-1) \times \frac{\Delta t - \Delta(t-1)}{\Delta(t+1) - \Delta(t-1)}, \tag{77}$$

where r_t is the interest rate which needs to be determined at time t with maturity Δ t and r(t-1), r(t+1) are the given short rates at time t-1, resp. t+1 with maturity $\Delta(t-1)$, resp. $\Delta(t+1)$. By interpolation we get following sequence of Euribor interest rates:

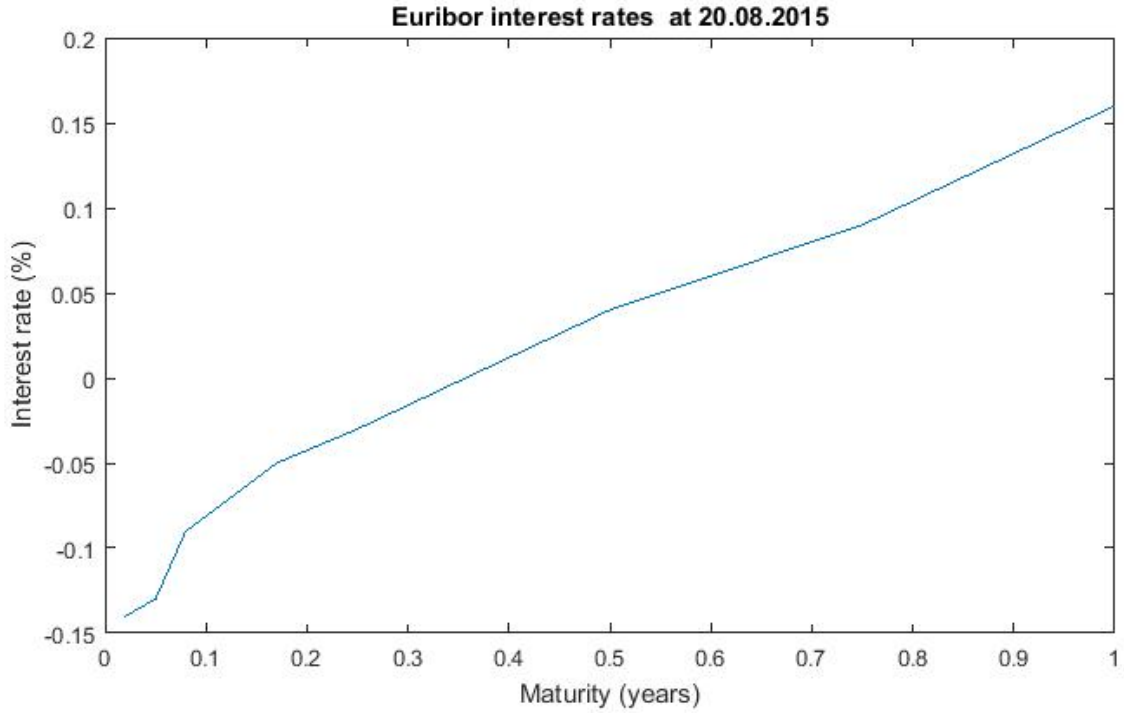

Figure 2: Interpolated Euribor interest rates on August 20, 2015.

Optimisation options

For the estimation of the Heston parameters we use the same strat values, same lower and upper bounds and the same termination conditions for all four optimisation methods. These are the optimal start parameters calculated from the historical bid/ask call data.

The given imput optimisation parameters are tested based on all traded bid and ask options on 20.08.2015. Out of sample testing are the traded options on 21.08.2015. The calculated optimal heston parameters are shown in Table 9

Table 7: Lower bounds, upper bounds and starting values of parameter estimation.

	kappa	theta	sigma	rho	v0
Lower Bounds	0.0001	0.0001	0.0001	-1	0.0001
Upper Bounds	100	10	10	1	10
Start	10.9	0.01	0.3	0.3	0.04

Table 8: Termination conditions

Maximum Function Evaluation	1,000
Maximum Iterations	20,000
Termination Tolerance on the Function Value	0.001

In sample testing

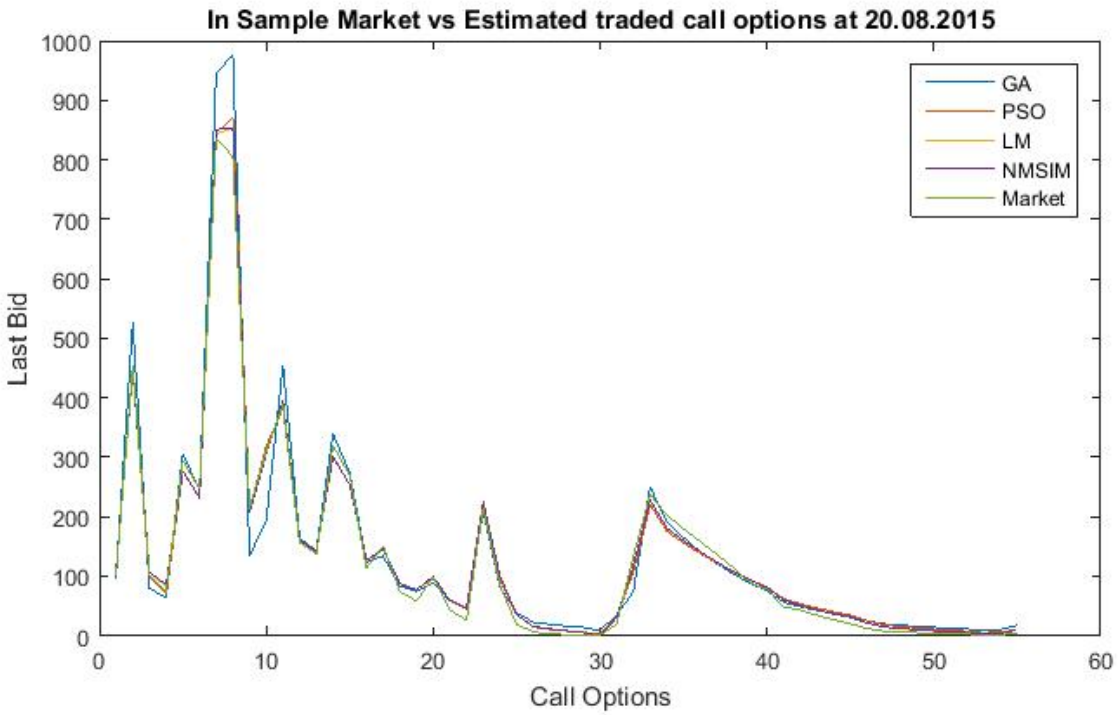

Figure 3: In-sample market vs. estimated traded call options at 20.08.2015.

Table 9: Example: Heston parameter estimation for 20.08.2015

	GA	PSO	LM	NMSim
kappa	0.0001	33.97239	11.93285	10.09133
theta	3.044759053	0.013567	0.014003	0.014909
sigma	0.626419737	0.0001	0.331356	0.382673
rho	-0.643085008	0.304603	-0.7736	-0.61471
v0	0.063062179	0.48146	0.200713	0.169579

Out of sample testing

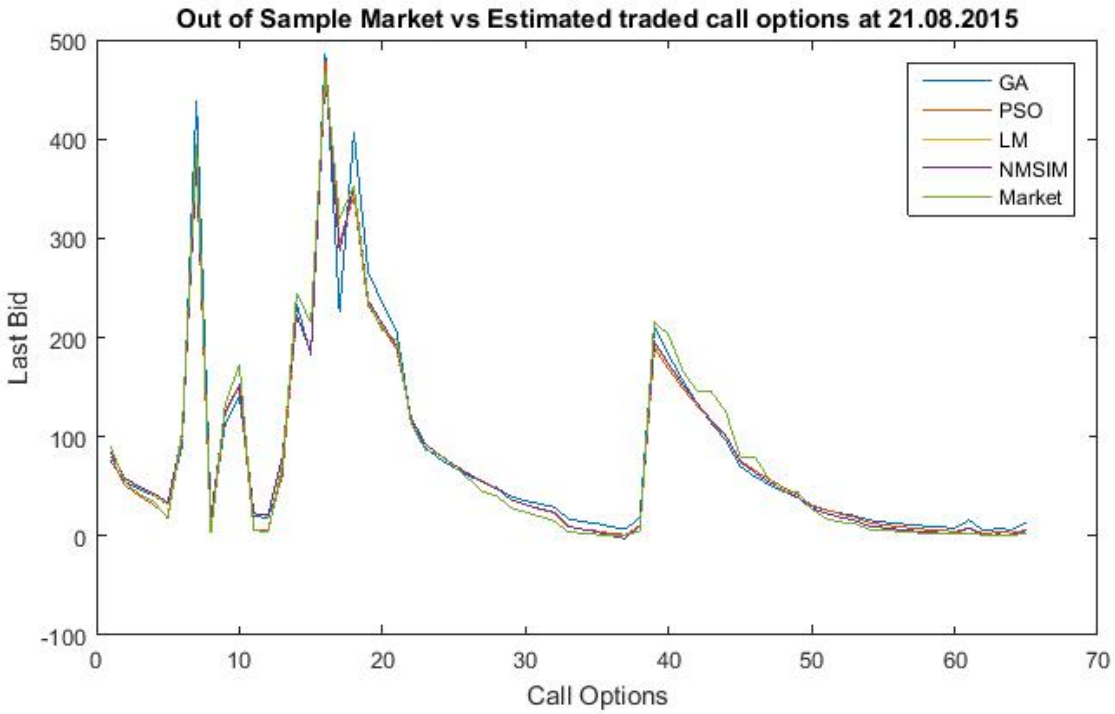

Figure 4: Out-of-sample market vs. estimated traded call options at 21.08.2015

Parameter estimation results by maturity

Table 10: RMSEs of parameter estimation by maturity in-sample.

Maturity	GA	PSO	LM	NMSim	LM+GA
0-3 Months	23.21	10.51	8.90	9.56	7.42
3-6 Months	20.99	11.73	9.14	9.74	8.21
6 Months - 1 Year	34.02	15.48	9.71	10.86	8.24
$>$ 1 Year	64.35	17.56	13.48	12.40	11.32

We can see that blocks of short maturities lead to smaller RMSEs in in-sample as well as out-of-sample estimations. In particular 0-3 months maturity outperforms longer dated maturities. For the estimation procedure we can see that LM+GA generates the smallest RMSE in this short maturity block in-sample as well as out-of-sample. In the the intermediate (3-6 months) maturities, LM+GA outperforms the

Table 11: RMSEs of parameter estimation by maturity out-of-sample.

Maturity	GA	PSO	LM	NMSim	LM+GA
0-3 Months	23.60	15.83	14.66	15.00	13.34
3-6 Months	26.14	17.09	15.16	14.90	15.03
6 Months - 1 Year	35.86	18.20	14.58	14.93	14.25
> 1 Year	68.58	27.23	24.16	28.31	23.15

other optimization methods just in-sample but not out-of-sample. Summarizing we can see a strong dominance of LM+GA method compared to the other optimization methods in-sample as well as out-of-sample.

Graph of in-sample (27.04.2015-30.12.2015) estimated parameters for a long maturity option with maturity 16.12.2016, Strike=3800:

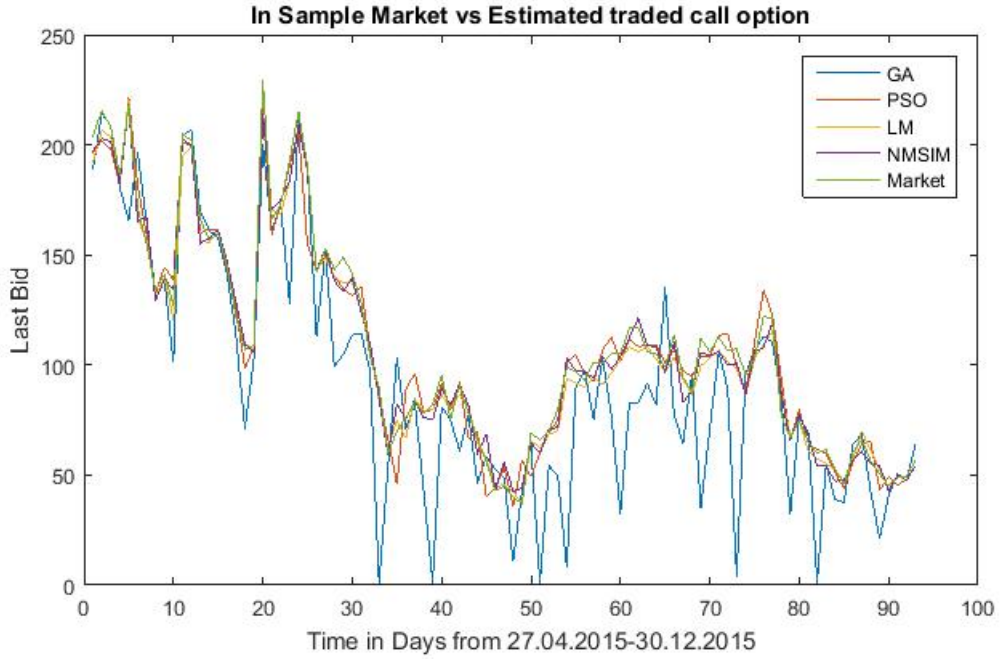

Figure 5: In-sample market vs. estimated traded call option with long maturity.

Graph of out-of-sample parameter estimation (28.04.2015-30.12.2015) for option with maturity 16.12.2016, Strike=3800 (Parameter are estimated using data 24 hours prior):

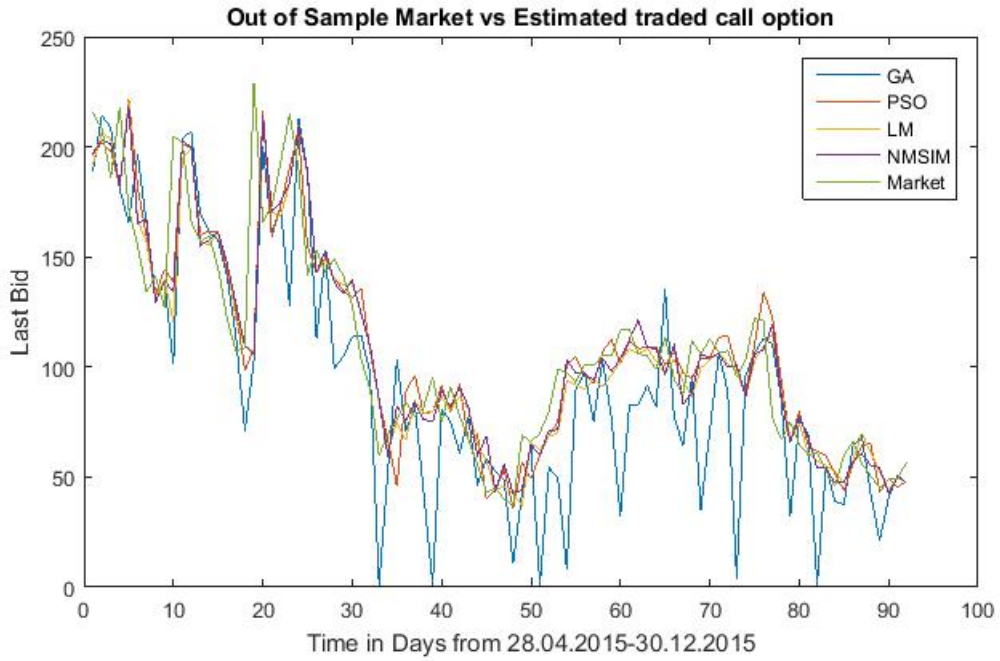

Figure 6: Out-of-sample market vs. estimated traded call option with long maturity.

Graph of in-sample (27.04.2015-30.12.2015) estimated parameters for a short maturity option with maturity 15.01.2016, Strike=3500:

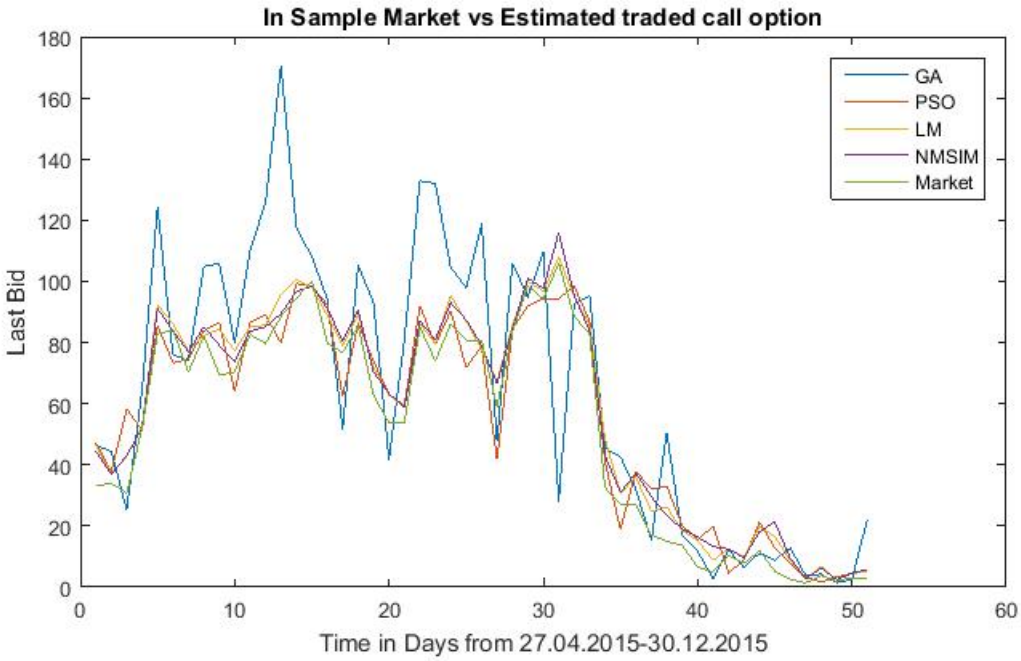

Figure 7: In-sample market vs. estimated traded call option with short maturity.

Graph of out-of-sample parameter estimation (28.04.2015-30.12.2015) for option with maturity 15.01.2016, Strike=3500 (Parameter are estimated using data 24 hours prior):

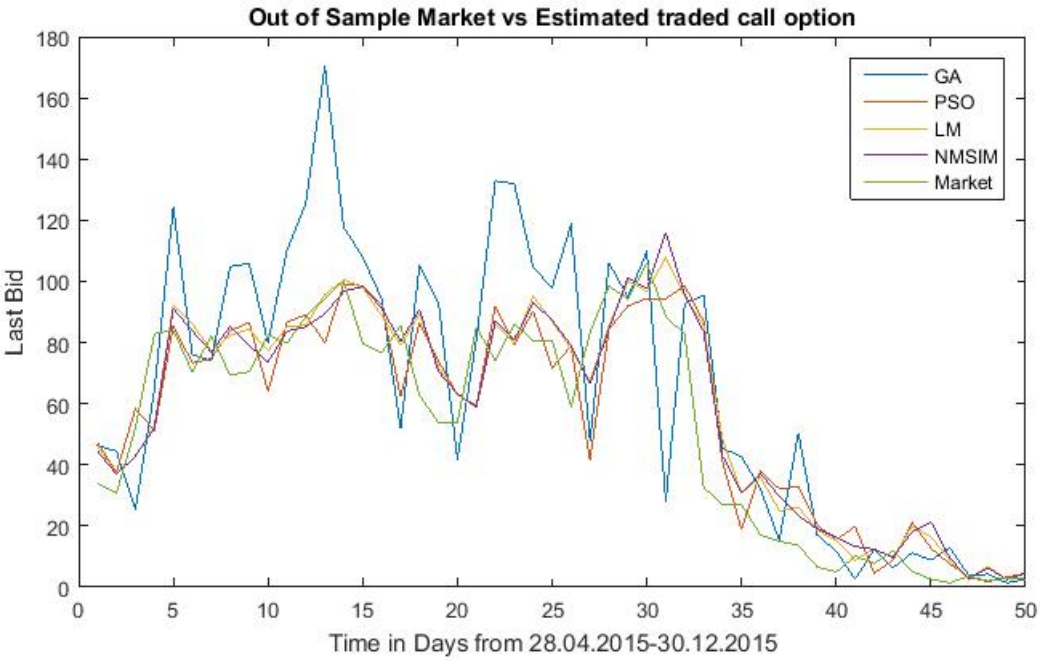

Figure 8: Out-of-sample market vs. estimated traded call option with short maturity.

Parameter estimation results by moneyness

Table 12: In-sample estimation-RMSE for 27.04.2015-30.12.2015

Moneyness	GA	PSO	LM	NMSim	LM+GA
At the money	49.00	18.98	12.59	13.80	11.52
Out of the money	45.53	11.90	8.00	8.20	7.95
In the money	53.84	27.78	22.41	21.62	22.04

Table 13: Out-of-sample estimation-RMSE for 28.04.2015-30.12.2015

Moneyness	GA	PSO	LM	NMSim	LM+GA
At the money	36.33	25.40	22.27	23.21	21.34
Out of the money	48.25	16.31	15.26	18.35	13.22
In the money	61.33	34.81	34.35	34.22	34.26

In-sample RMSEs are smallest using the LM+GA method for at-the-money as well as out-of-the-money options. Using in-the-money option we can see that the

NMSim -method produces the smallest RMSE in-sample. We get similar results for the out-of-the-sample test which indicates that LM+GA works best when using at-the-money as well as out-of-the-money options for parameter estimation, whereas NMSim generates the smallest RMSE using in-the-money options for parameter estimation. Since LM+GA provides the strongest test results we apply this combination for the calibration of the Heston parameters.

Moreover, we determine the optimal time frame for historic parameter estimation. From all these combinations we then rank the options due to their best calibration results. Our tests show that the optimal Heston parameters can be calculated using data from the best 23 options of the last 5 days, selected by maturity and moneyness.

We next present the corresponding graphs using the LM+GA method.

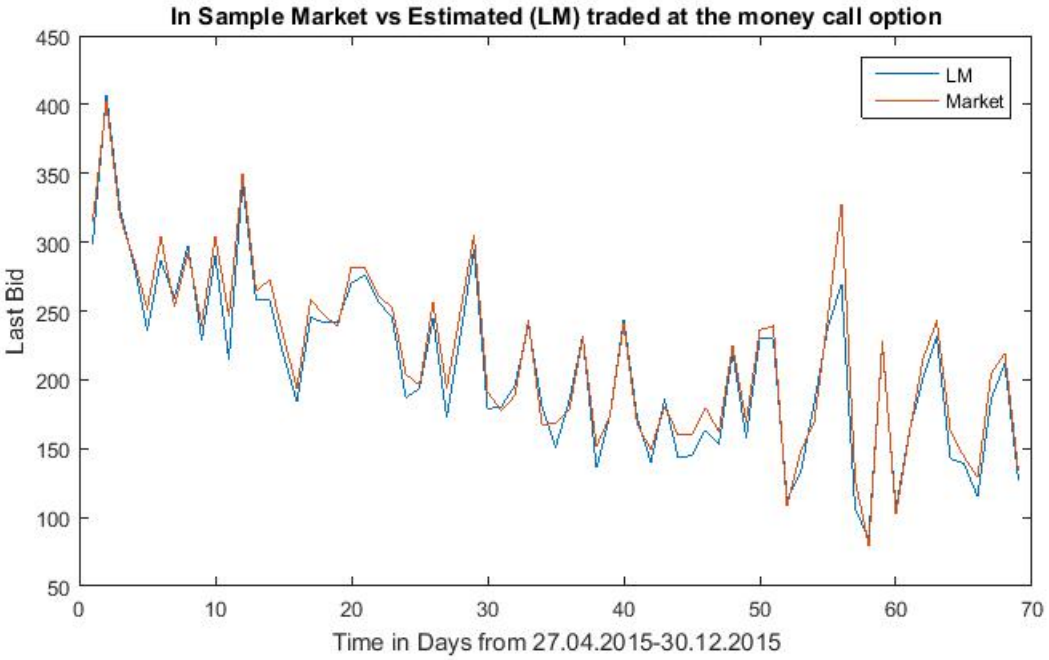

Figure 9: In-sample market vs. estimated traded call option at-the-money call option.

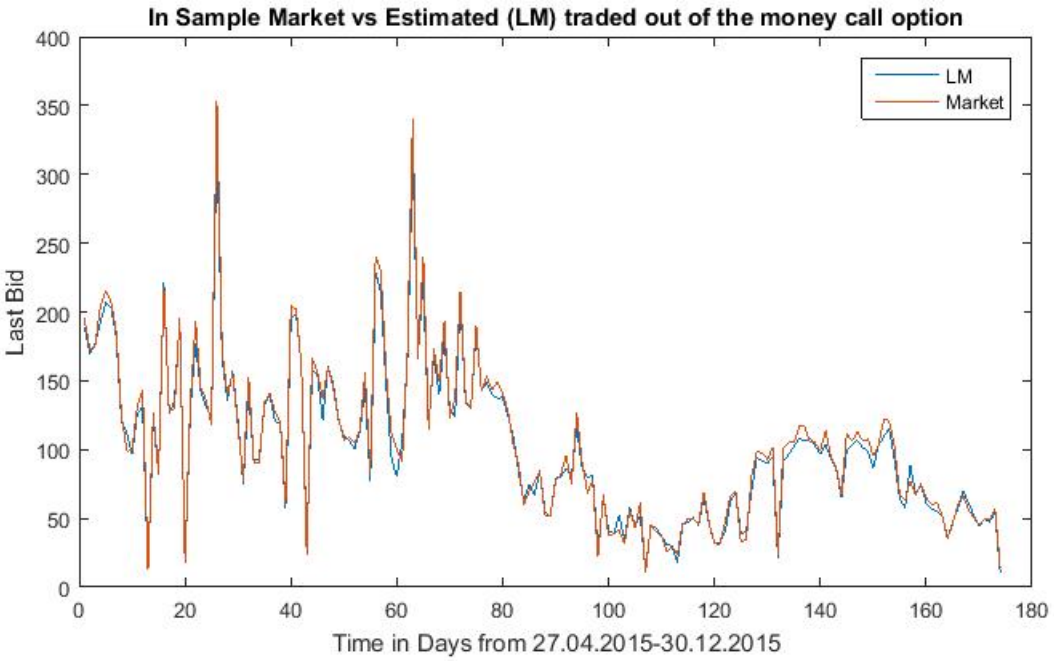

Figure 10: In-sample market vs. estimated traded call option out-of-the-money call option.

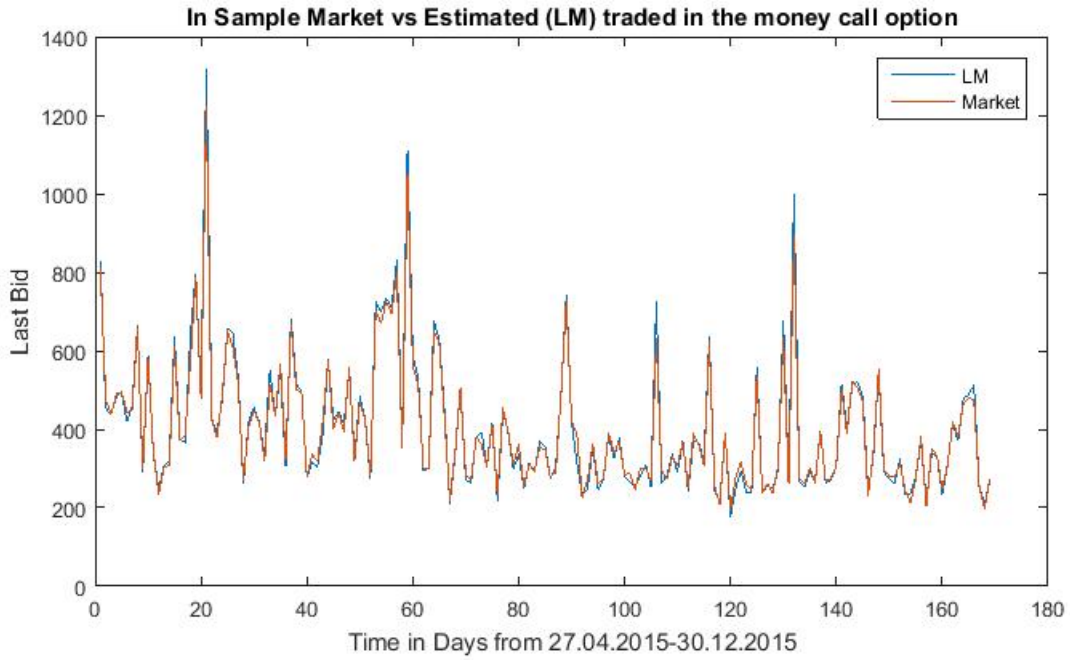

Figure 11: In-sample market vs. estimated traded call option in-the-money call option.

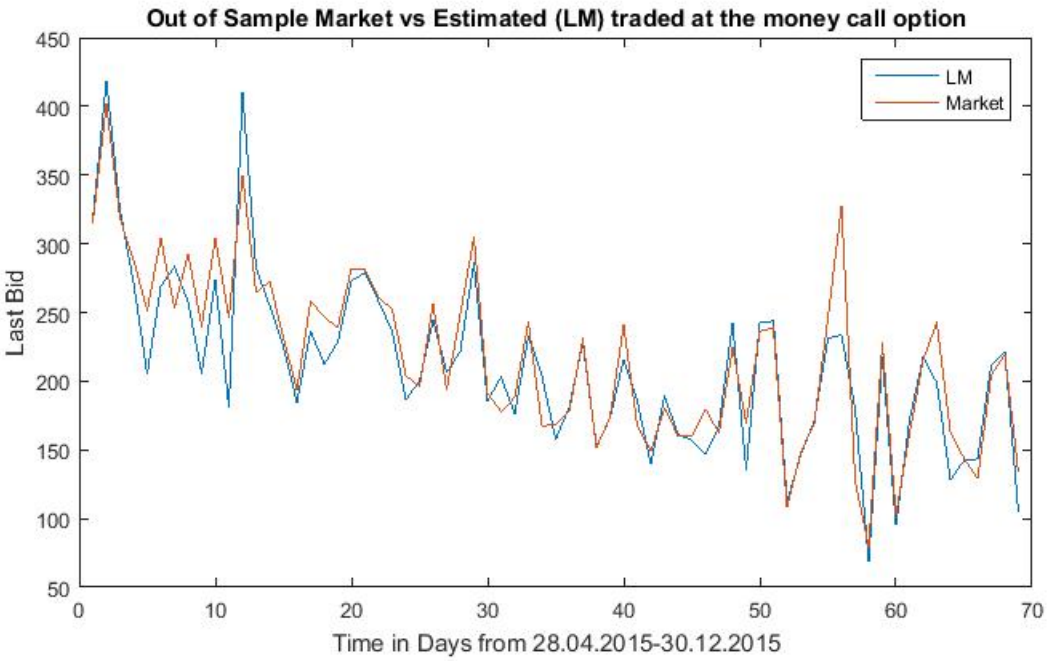

Figure 12: Out-of-sample market vs. estimated traded call option at-the-money call option.

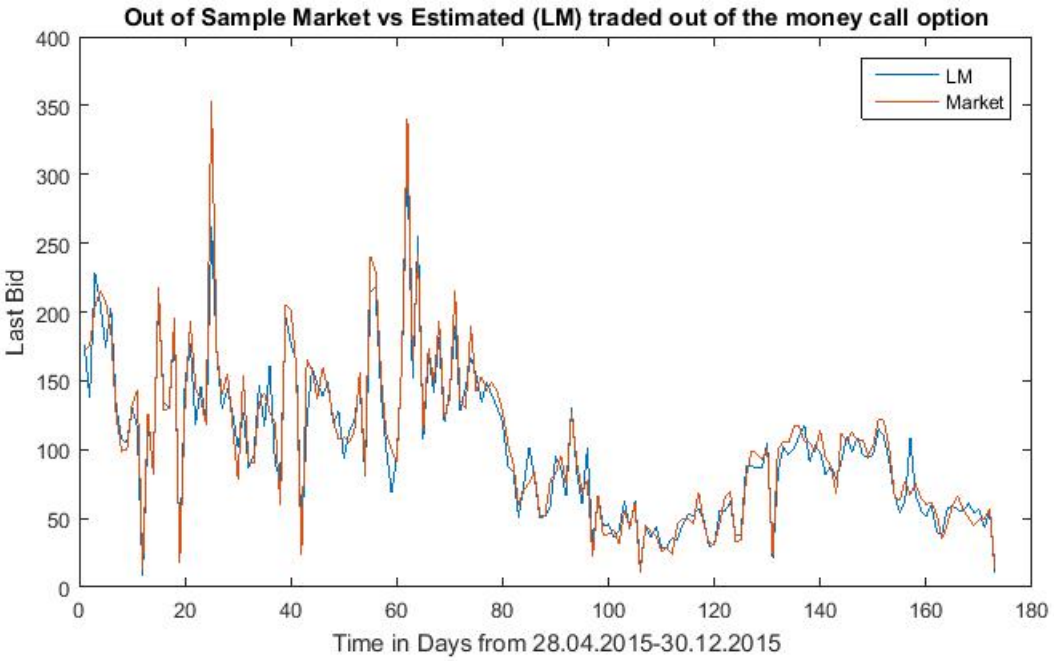

Figure 13: Out-of-sample market vs. estimated traded call option out-of-the-money call option.

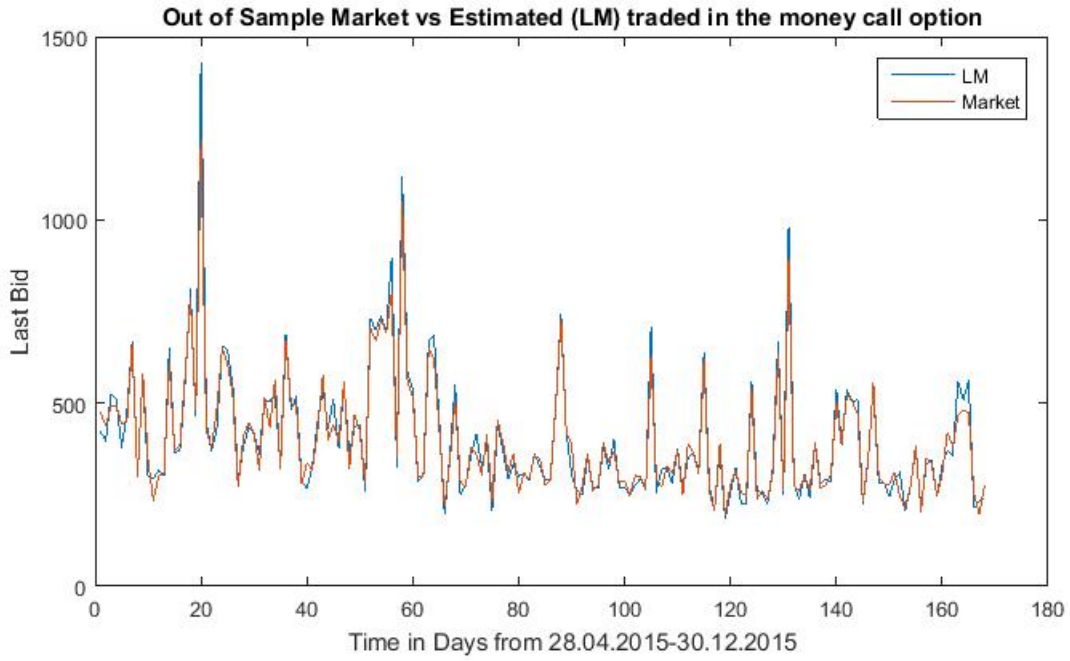

Figure 14: Out-of-sample market vs. estimated traded call option in-the-money call option.

RMSE of all options

Table 14: Estimation RMSEs of all options 27.04.2015-30.12.2015

	GA	PSO	LM	NMSim	LM+GA
In Sample	46.85	15.30	11.27	11.85	10.23
Out of Sample	82.78	29.51	19.25	22.45	17.36
Computational Time in Seconds	38.106	61.248	2.610	5.742	42.987

The general test results show the smallest RMSE for the LM method which indicates that LM is superior to the other optimization methods. Not only that LM generates the smallest RMSE, it is also the fastest optimization method in terms of computational time.

Even though LM yields the best calibration results among all optimization methods, we can achieve an improvement in the test results by combination of the LM method and Genetic Algorithms.

2.4.2 Validation of Chi-Square optimization

For validation of the Chi-Square test results we take historic FESX50 tick data and accumulate all tick changes, regardless of price or volume changes or new order submission, at each second. We take tick data from April 26, 2015 to Dec 30, 2015 and test in-sample as well as out-of-sample. The in-sample test comprises 1-50 days, whereas the out-of-sample test takes these 1-50 in-sample days as estimation set and tests 1 day out-of-sample.

Table 15: In sample Chi-Square test results for Compound Poisson CDF.

Nmbr. of Days	λ	χ^2	1-p	sign. level	H0
1	252.69	18,224.5	0.00	0.05	rejected
2	249.09	34,218.95	0.00	0.05	rejected
3	250.71	25,293.35	0.00	0.05	rejected
4	248.54	36,186.20	0.00	0.05	rejected
5	249.74	44,040.30	0.00	0.05	rejected
10	246.60	25,882.64	0.00	0.05	rejected
20	242.14	24,100.07	0.00	0.05	rejected
50	238.20	31,618.24	0.00	0.05	rejected

Table 15 shows that H_0 for Compound Poisson distribution can be rejected for

all tested in-sample time lengths. The real observed order submission arrival times cannot be described in-sample by a Possion distribution.

Table 16: Out-of-sample Chi-Square test results for Compound Poisson CDF.

Nmbr. of Days	λ	χ^2	1 - p	sign. level	H0
1	266.39	38,640.16	0.00	0.05	rejected
2	266.39	42,448.99	0.00	0.05	rejected
3	266.39	46,458.58	0.00	0.05	rejected
4	266.39	64,251.93	0.00	0.05	rejected
5	266.39	62,840.45	0.00	0.05	rejected
10	266.39	31,743.15	0.00	0.05	rejected
20	266.39	53,555.67	0.00	0.05	rejected
50	266.39	70,273.06	0.00	0.05	rejected

Table 16 shows that H_0 for Compound Poisson distribution can be rejected for all tested out-of-sample time lengths. The real observed order submission arrival times cannot be described out-of-sample by a Possion distribution.

Table 17: In sample Chi-Square test results for inverse transformation.

Nmbr. of Days	λ	χ^2	1-p	sign. level	H0
1	252.69	974.55	0.70	0.05	accepted
2	249.09	926.65	0.95	0.05	accepted
3	250.71	1,044.85	0.15	0.05	accepted
4	248.54	1,016.62	0.34	0.05	accepted
5	249.74	968.73	0.75	0.05	accepted
10	246.60	1,047.94	0.14	0.05	accepted
20	242.14	999.89	0.49	0.05	accepted
50	238.20	1,042.93	0.16	0.05	accepted

Table 17 shows that H_0 for inverse transformation can be accepted for all tested in-sample time lengths, meaning that a real observed order submission arrival times can be described in-sample by a inverse transformation.

Table 18: Out sample Chi-Square test results for inverse transformation.

Nmbr. of Days	λ	χ^2	1-p	sign. level	H0
1	266.39	16,446.98	0.25	0.05	accepted
2	266.39	22,647.4	0.57	0.05	accepted
3	266.39	23,652.11	0.00	0.05	rejected
4	266.39	27,027.48	0.01	0.05	rejected
5	266.39	25,916.66	0.48	0.05	accepted
10	266.39	32,191.94	0.00	0.05	rejected
20	266.39	38,135.89	0.21	0.05	accepted
50	266.39	56,267.96	0.00	0.05	rejected

Table 18 shows that H_0 for inverse transformation can be accepted for some tested out-of-sample time lengths, meaning that a real observed order submission arrival times can partly be described out-of-sample by a inverse transformation.

Summarizing, the best estimation result can be achieved by the choice of inverse transformation and the selection of a 2 day time window.

The graphical representation of the Compound Poisson PMFs shows that the historical bid and ask order submission volumes are not Compound Poisson distributed whereas the inverse transformation indicates very good Chi-Square test results.

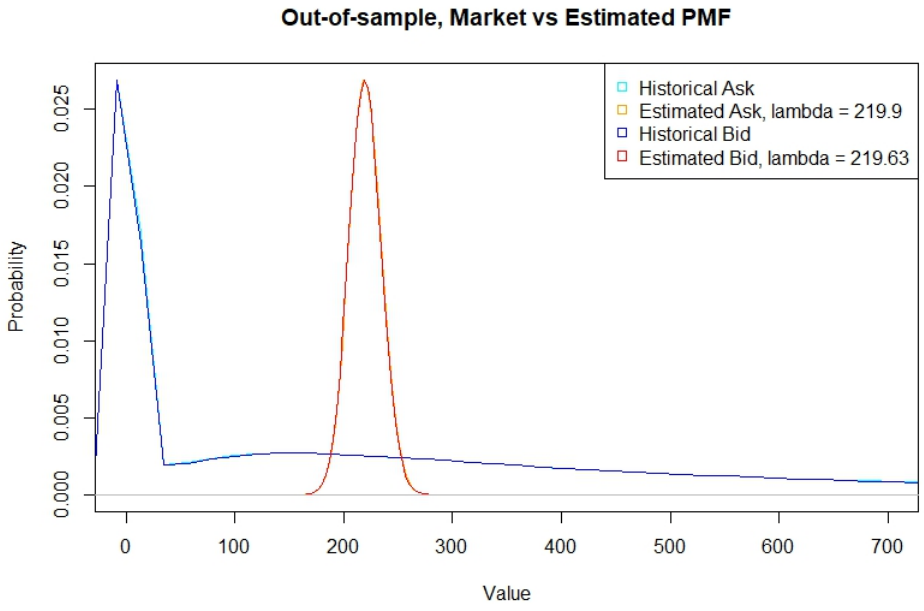

Figure 15: Out-of-sample test results: Market vs. Estimated ask order volume submission Compound Poisson CDF.

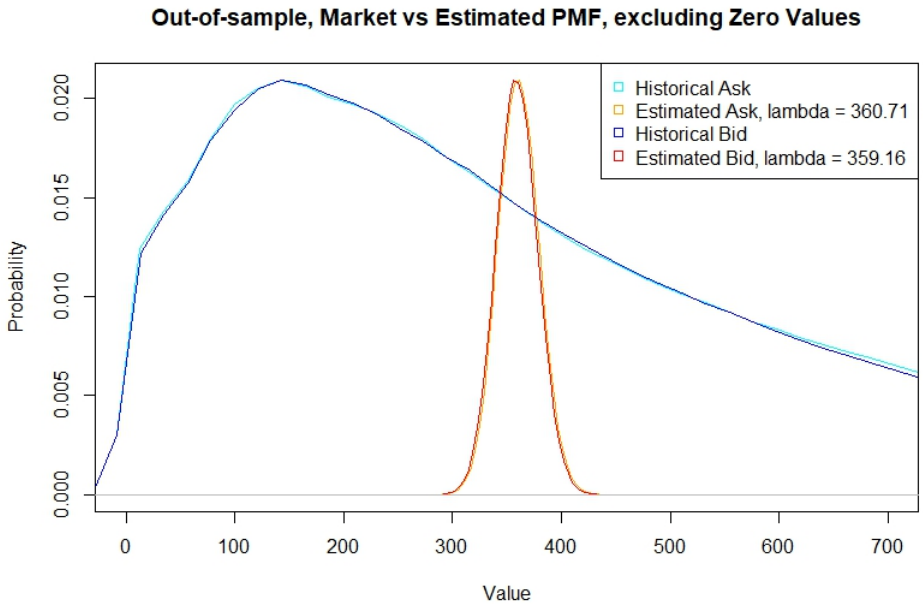

Figure 16: Out-of-sample test results: Market vs. Estimated ask order volume submission Compound Poisson CDF without zero values.

As we can see in Figure 15 and 16, the histogram reveals that there are many seconds where there are no changes or new submissions in bid or ask orders. Therefore we can see many zeros in our test sample. In order to avoid distortions of the Compound Poisson process we also test for Compound Poisson process without these zero values.

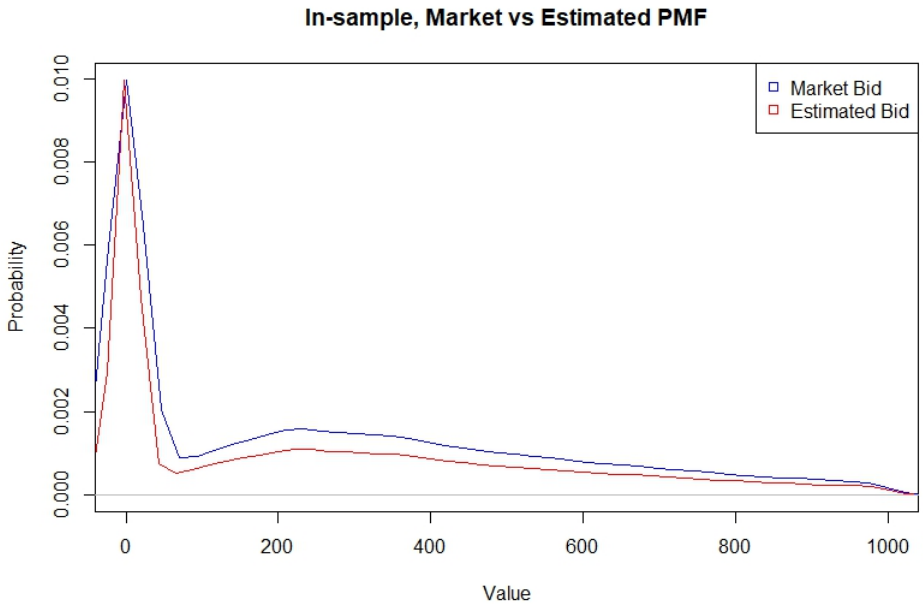

Figure 17: In-sample test results: Market vs. Estimated bid order volume submission inverse transformation.

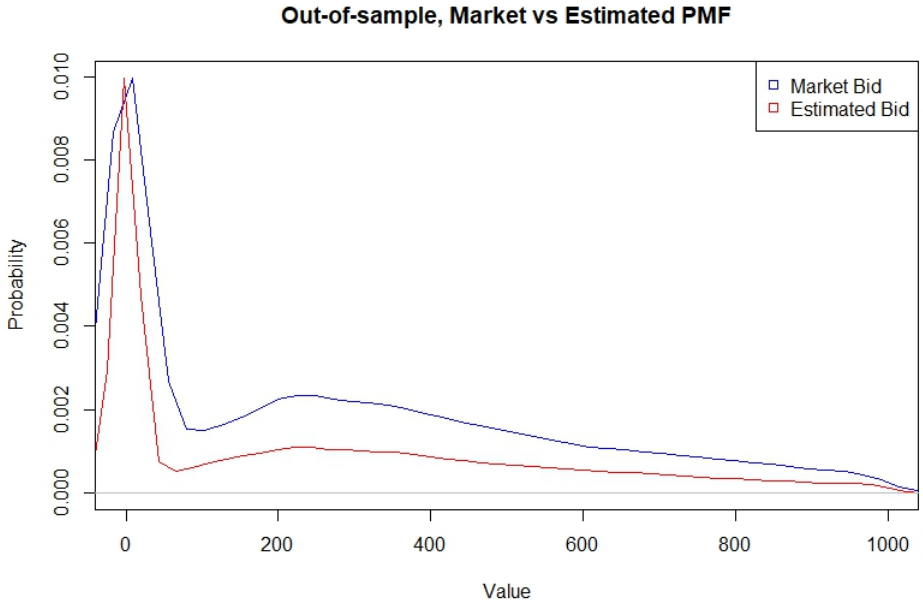

Figure 18: Out-of-sample test results: Market vs. Estimated bid order volume submission inverse transformation.

2.4.3 Validation of liquidity estimation

After application of the different optimization methods we take a look at the results and compare it to the results which emerge of the Black-Scholes framework (BS), where we estimate the liquidity values at each minute, due to computational time restrictions. This setup is usually used for simulating price processes by assuming a complete market. In the Heston framework we have to deal with an incomplete pricing world which leads us to the necessity of adding a contingent claim to the replicating portfolio. Therefore BS serves as a benchmark for us. For each optimization technique we want to conduct the calibration procedure for estimation of the future market liquidity and compare it to the estimation quality when using standard Brownian Motion for simulation of the bid and ask prices, as well as the traded volumes. This comparison gives us just a relative measure of quality. Therefore we also compare the estimated market liquidity to the realized one. For this purpose we perform the market liquidity estimation for each optimization method.

Table 19: RMSEs of Heston and GBM model for liquidity estimation.

Heston	2.2836
GBM	3.2228

We can see that Heston model outperforms the GBM model. This indicates that stochastic volatility is an important factor in market liquidity modeling. Since GBM assumes a constant volatility we can see that by using a powerful optimization method for parameter estimation we can generate more realistic results when modeling market liquidity.

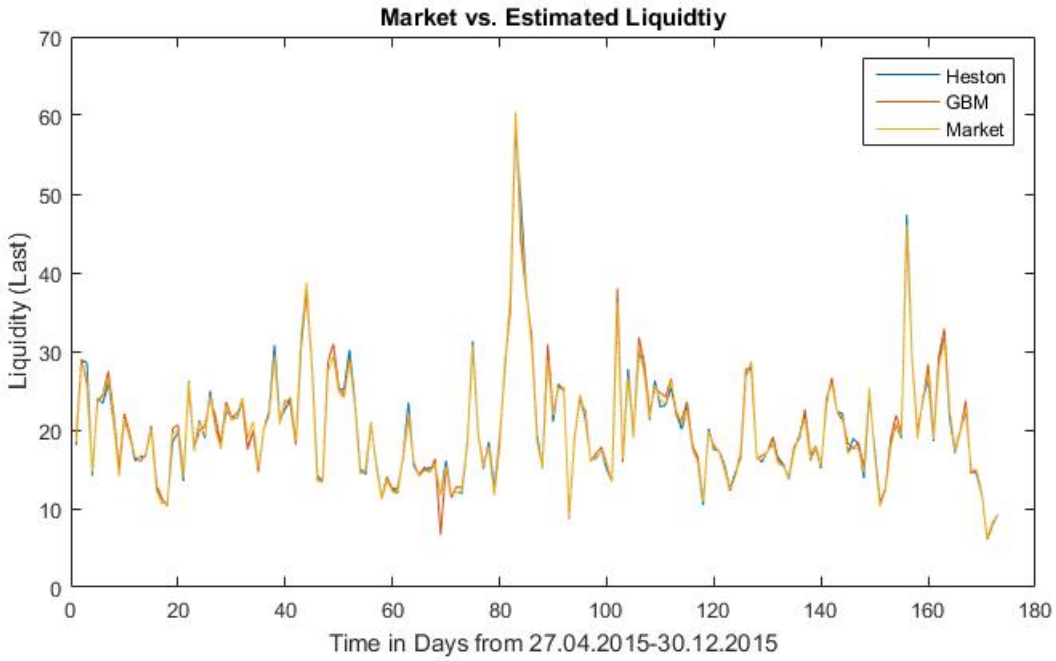

Figure 19: Comparison of realized market liquidity (yellow) vs. estimated market liquidity with Heston model (blue) and GBM model (red) from 27.04.2015-30.12.2015.

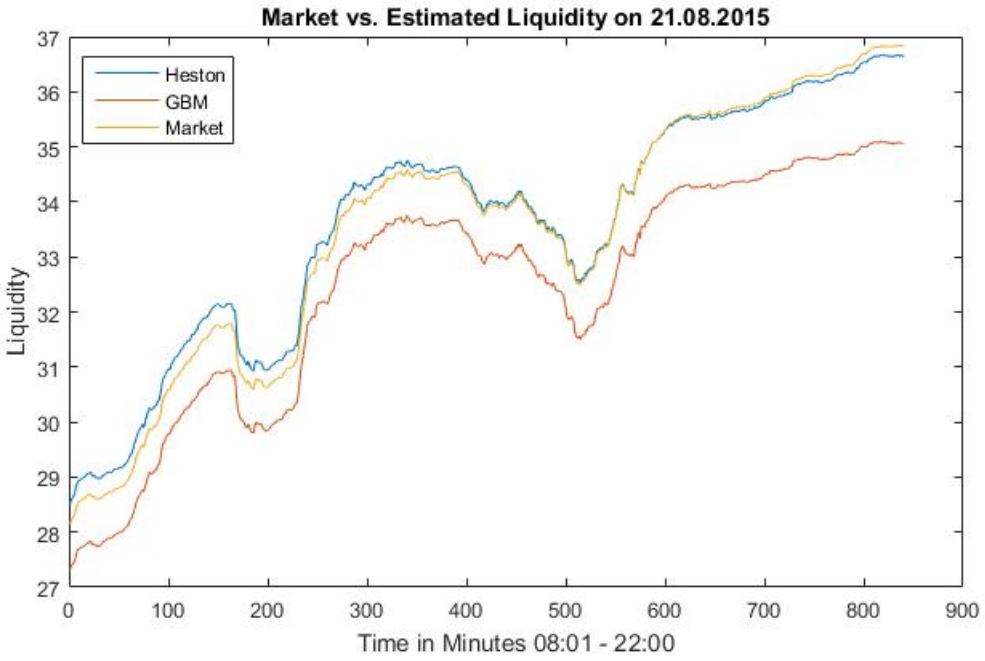

Figure 20: Comparison of realized market liquidity (yellow) vs. estimated market liquidity with Heston model (blue) and GBM model (red) on 21.08.2015.

2.5 Conclusion

This work compares the estimation power of the Heston model and the Geometric Brownian motion (GBM) model as well as Inverse transformation sampling and Compound Poisson proces. The application is a unique approach for market liquidity estimation.

For the simulation of bid and ask prices we perform Heston und GBM Model. To calibrate the Heston parameters we propose several optimization methods and compare their estimation power in-sample as well as out-of-sample by taking historic data of Euro Stoxx 50 Future options. The estimation procedures Heston parameters are the Genetic algorithms (GA), the Particle Swarm Optimization technique (PSO), the Levenberg-Marquardt method (LM) and the Nelder-Mead Simplex method (SM). We find a superior estimation power of the LM method compared to the other optimization techniques. We test for different option maturities as well as for different moneyness levels and find a higher estimation power in shorter dated maturities. The LM method works best for Out-of-the money and at-the-money options whereas the SM method works best for in-the-money options. Generally the LM method does not only generate the smallest RMSE, but also performs best in

computational time.

For the simulation of the bid and ask volume, we apply two types of processes: Compound Poisson process and Inverse transformation sampling. For a 2 day sampling window, the Chi-Square test indicates a superiority of the Inverse transformation sampling compared to the Compound Poisson process, since the historical bid and ask order submission volumes are not Compound Poisson distributed. By using the Inverse transformation sampling, the forecasting error is significantly lower than with simulated volumes using the Compound Poisson process. This indicates that the best choice for the simulation of bid and ask volumes is the Inverse transformation sampling method.

For liquidity estimation this means, that by simulation of the bid and ask prices, generated by the calibrated heston model, it is possible to estimate market liquidity up to 29.14% better than with GBM model. The results are robust for in-sample as well as out-of-sample tests. The only drawback of using the Heston model is the high computation time, necessary for the calibration of the parameters and simulation of the prices.

References

[1] H.H.N. Amin, Calibration of Different Interest Rate Models for a Good Fit of Yield Curves, Thesis, *Delft University of Technology Faculty of Electrical Engineering*, Mathematics and Computer Science Delft Institute of Applied Mathematics, September 2012.

[2] Monika Piazzesi, Affine Term Structure Models, *Handbook of Financial Econometrics*, 2010, Vol. 1, Chapter 12, 697-700.

[3] John Y. Campbell, A Defense of Traditional Hypotheses about the Term Structure of Interest Rates, *The Journal of Finance*, Vol. 41, No. 1 (Mar., 1986), pp. 183-193.

[4] Y.-S. Kim, Refined Simplex Method for Data Fitting, *Astronomical Data Analysis Software and Systems VI*, ASP Conference Series, Vol. 125, 1997.

[5] ETF Specific Data Point Methodologies, Morningstar,*Morningstar Methodology Paper*, December 31, 2010.

[6] Kennedy J. and Eberhart R., Particle Swarm Optimization, *Proceedings of the Fourth IEEE International Conference on Neural Networks*, Perth, Australia. IEEE Service Center(1995) 1942-1948.

[7] Kujtim Avdiu and Stephan Unger, Secondary hedging for contingent claims, *Journal of Financial and Economic Practice*, Vol 13, issue 2, 2013.

[8] Marquardt, D. W., An Algorithm for Least Squares Estimation of Nonlinear Parameters, *Journal Society Industrial Applied Mathematics*, Vol. 11, No. 2, pp. 431-441, June 1963.

[9] S.C.Wong, C.K.Wong and C.O.Tong, A parallelized genetic algorithm for the calibration of Lowry model, *Parallel Computing*, 27: 1523-1536, 2001.

[10] Evelyn Araneda, A Variation of the Levenberg Marquardt Method. An attempt to improve efficiency. *Submitted to the Department of Earth, Atmospheric and Planetary Sciences in Partial Fulfillment for the Master of Science in Geosystems*, MIT, May, 2004.

[11] Jörg Kienitz and Daniel Wetterau, Financial modelling, Theory, Implementation and Practice with Matlab Source, *John Wiley & Sons Ltd*, p. 437, 2012.

[12] Ingber, A. L. (1995), Adaptive simulated annealing (asa): Lessons learned, Control and Cybernetics.

[13] Kirkpatrick, S., Jr., C. D. G. and Vecchi, M. P. (1983), Optimization by simulated annealing, Science 220,(4598), 671-680.

[14] Mikhailov, S. and Nögel, U. (2003), Hestons stochastic volatility model implementation, calibration and some extensions, Wilmott.

[15] Gatheral, J. (2004), Lecture 1: Stochastic volatility and local volatility, Case Studies in Financial Modelling Notes, Courant Institute of Mathematical Sciences.

[16] Heston, S. L. (1993), A closed-form solution for options with stochastic volatility with applications to bonds and currency options, The Review of Financial Studies 6 (2), 327-343.

[17] Rudolf Bauer, Fast Calibration in the Heston Model, Master Thesis, *Vienna University of Technology*, 15 February 2012.

[18] Seungho Yang, Hyejin Park, Jaewook LeeRobust, Calibration of the Stochastic Volatility Model, *Preprint submitted to Elsevier*, Department of Industrial and Management Engineering Pohang University of Science and Technology, South Korea, 23 April 2010.

[19] Stephan Unger and Lane P. Hughston, Stochastic Liquidity with Conditional Volatility, *Working paper*, Vienna University of Economics and Business, Imperial College London, 2010.

[20] Stephan Unger and Lane P. Hughston, Market Liquidity Measurement, *Working paper*, Vienna University of Economics and Business, Imperial College London, 2010.

[21] Kuok King Kuok and Chiu Po Chan, Particle Swarm Optimization for calibrating and optimizing Kinjiang model parameters, *International Journal of Advanced Computer Science and Application*, 2012, Volume 3, Nr. 9, p.115-123.

[22] Manfred Gilli and Enrio Schumann, Calibrating option pricing model with heuristics, *COMISEF Working paper series*, WPS-030, March 8, 2010.

[23] Warrick Poklewski-Koziell, Stochastic volatility models: Calibration, pricing and hedging, *University of the Witwatersrand*, Dissertation, May 2012.

[24] Mark Lu, Liang Zhu, Li Ling, Gary Zhang, Walter Chan and Xin Zhou, Distributed model calibration using Levenberg-Marquardt algorithm, *Proc. of SPIE*, 2007, Volume 6520, 65302C-1-8.

[25] A calibration method for HJM models based on the evenberg-Marquardt optimization algorithm, *Politecnico Di Milano*, 2009-2010.

[26] Stephan Unger and Lane P. Hughston, Market Liquidity Measurement, *Working Paper*, Vienna University of Economics and Business Administration, October 2011.

[27] Rudolf Bauer, Fast Calibration in the Heston Model, Master Thesis, *Vienna University of Technology*, 15 February 2012.

[28] Seungho Yang, Hyejin Park, Jaewook LeeRobust, Calibration of the Stochastic Volatility Model, *Preprint submitted to Elsevier*, Department of Industrial and Management Engineering Pohang University of Science and Technology, South Korea, 23 April 2010.

[29] Peter Carr and Dilip B. Madan, Option valuation using fast Fourier Transform, *Journal of Computational Finance*, 2:61-73, 1999.

[30] Wolfgang Putschögl, On Calibrating Stochastic Volatility Models with time-dependent Parameters, *arXIV:1010.1212v1*, qfi.PR, 6 October, 2010.

[31] D.E.Goldberg, Genetic Algorithms in Search, optimization and Machine Learning, *Addison-Wesley*, Reading, MA, 1989.

[32] Spall, J.C., Introduction to Stochastic Search and Optimization, *Wiley-Interscience Series in Discrete Math and optimization*, 2001.

[33] Evelyn Araneda, A Variation of the Levenberg Marquardt Method. An attempt to improve efficiency. *Submitted to the Department of Earth, Atmospheric and Planetary Sciences in Partial Fulfillment for the Master of Science in Geosystems*, MIT, May, 2004.

[34] Henri P. Gavin, The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems, *Department of Civil and Environmental Engineering*, Duke University, September 24, 2013.

[35] Marquardt, D. W., An Algorithm for Least Squares Estimation of Nonlinear Parameters, *Journal Society Industrial Applied Mathematics*, Vol. 11, No. 2, pp. 431-441, June 1963.

[36] Ruiz, Andres N.; Cora, Sofia A.; Padilla, Nelson D.; Dominguez, Mariano J.; Vega-Martinez, Cristian A.; Tecce, Tomas E.; Orsi, Alvaro; Yaryura, Yamila; Garcia Lambas, Diego; Gargiulo, Ignacio D.; Munoz Arancibia, Alejandra M., Calibration of Semi-analytic Models of Galaxy Formation Using Particle Swarm Optimization, *Published 2015 March 12, 2015 The American Astronomical Society*

[37] Roland Lichters, Markus Trahe, Efficient Simulation of the Multi Asset Heston Model, *Published January 2016, SSRN Electronic Journal*

3 Implicit Hedging and Liquidity Costs of Structured Products

Kujtim Avdiu [1] and Stephan Unger[2]

[1] OeNB, Otto-Wagner-Platz 3, 1090 Vienna, Austria
e-mail: kujtim.avdiu@univie.ac.at

[2] Department of Economics & Business
Saint Anselm College, New Hampshire, USA
e-mail: sunger@anselm.edu

July 2020

Abstract

This article analyzes the implicit hedging and and liquidity costs of structured equity products offered by various financial institutions. We replicate several payoffs of structured products and compare the calculated fair value, based on Heston model, using various optimization techniques and compare their fair values with the historic prices traded in the market. We find that implicit hedging costs range between 0.9% to 2.9% markup on the fair value, where we find the underlying market volatility to be the relevant driver of this range for complex structures, while market liquidity can be extracted as the only driver of markups for simple structures with no hedging requirements.

Keywords: Pricing, Heston model, hedging costs, structured products, optimization.

Author contribution
K.A. developed the idea to investigate the markup between the fair value and real-traded prices of structured products. S.U. added the idea to test the statistical coherence between market volatility, resp. market liquidity and this markup. K.A.was responsible for data collection and calculations. S.U. & K.A. interpreted the results together and wrote the conclusion.

Submitted: Journal of alternative Investment

3.1 Introduction

Structured products are customized financial products which are composed of plain vanilla financial products and derivatives. They can be categorized into leverage, participation, return optimization and capital guarantee products. Each category exhibits different markups on their fair value due to different market conditions, such as market volatility and market liquidity, and implicit risk factors, such as the payoff of the structured product. While the detection of structured products markup has widely been proved by literature, the drivers of these markups have not been investigated extensively. Moreover, what has been neglected by literature so far, is the exploration of the drivers of the changes in this markup over time. Since there exist several reasons for issuers of structured products to charge a markup, the question remains what determines the change in the markup over time in the secondary market.

[Henderson and Pearson(2011)] analyze offering prices of 64 issues of a popular retail structured equity product and find that offered prices are almost 8% greater than estimates of the products' fair market values obtained using option pricing methods.

[Burth et al.(2016)] distinguish between distinguish convex strategies, which convey a payoff resembling a long position in a stock portfolio together with a protective put, and concave strategies which replicate covered call payoffs. They examine 275 concave products sold in the Swiss market in the late 1990s, and compare their prices to an estimate of the cost of creating the payoffs using options traded on EUREX. Not surprisingly, they find that prices of the concave products seem to be quite favorable for the banks. They also find considerable pricing dispersion, with distinct differences across different issuing institutions, between products that pay a coupon versus those that do not, and between instruments issued by a single bank versus those with co-lead managers.

[Stoimenov and Wilkens(2005)] examine the pricing of equity-linked structured products in the German market. They compare the closing prices of 2566 equity-linked structured products on the German stock index DAX (Deutscher Aktienindex) to theoretical values derived from the prices of options traded on the Eurex (European Exchange) and find that at issuance, structured products on DAX stocks sell at an average of 3.89% above their theoretical values based on Eurex options.

[Bergstresser(2008)] presents evidence on the abnormal returns of a broad sample of SEPs that is consistent with the findings in literature of overpricing at the offering date, such as by [Rogalski and Seward(1991)] and [Jarrow and O'Hara(1989)].

[Wilkens et al.(2003)] compare the daily closing quotes of roughly 170 reverse convertibles and 740 discount certificates to values based on duplication strategies during November 2001, using call options traded on the Eurex (European Exchange).

They investigate the average price differences dependent on product type, issuer, and underlying. They put a special focus on the possible influence of order flow, i.e. they analyze, using product life cycles and moneyness as proxies, whether the price quotes depend on the expected volume of purchases and sales. The study reveals significant differences in the pricing of structured products, which can mostly be interpreted as being in favor of the issuing institution.

So far, mostly driving factors explaining the divergence at issuance have been investigated. As the fair value of a structured product fluctuates over time, meaning that the markup is not constant after issuance of the product, the drivers for these changes have not been investigated so far. In assessing the driving factors of pricing policies, [Wilkens et al.(2003)] conclude that issuers orient their pricing towards the product lifetime and the incorporated risk of a redemption by shares (given by the moneyness of the implicit options), bearing in mind the volumes of sales and repurchases to be expected from issuance until maturity.

We assess the driving factor for the changes in the markup over time by setting up the following hypothesis:

(H1): The changes in the underlying volatility explain the changes in markup.
(H2): The changes in the underlying liquidity explain the changes in markup.

Moreover we are the first to our knowledge who apply the Heston model for the calculation of fair value over the entire test period and thus for investigating the divergence between fair value and traded price of a structured product. We replicate the payoff of the structured products on several clusters using parallel computing techniques. The replication using cluster technology represents a novelty in the pricing of structured products as it allows more simulations to approximate closer the fair value of these products.

Since several studies find the markup to be non-zero, due to incurred hedging costs, we investigate why the the markup is not of constant size, but what drives the markup changes over time. We are interested in finding the reason for these fluctuations. One possible explanation might be the dependency of hedging costs on underlying market volatility or liquidity. Therefore, we regression of structured products markup on market volatility and liquidity. Moreover, we replicate complex products, and not standard products such as classic, rainbow, guarantee, turbo, and barrier products, but combination of all these products, by using an optimal calibrated Heston model in order to calculate a robust fair value over time.

We structure this article as follows: In section 3.2 we give an overview about the data and methodology we use. In section 3.3 we present the structured products we replicate. In section 3.4 we price all these products for the past X years, using the calibrated Heston model, in order to be able to compare them with real traded

market prices. The difference between these two prices reveals the markup, thus the implicit hedging costs. In section 3.5 we conduct the regression to test if market volatility, resp. the changes in market volatility, explains the size of the markup, resp. its changes. In section 3.6 we present the results obtained and we finally conclude.

3.2 Data and methodology

Subject of investigation are equity products due to better data availability of historic equity option prices for calibration of the Heston model. The structured products we replicate are the following:

1. Dual Index kick-out, Capital risk
2. Dual Index + Coupon, Capital risk
3. Triple Index + Coupon, Capital risk
4. Single Index + Coupon, Capital risk
5. Single Index + maturity Coupon, Capital protection
6. Single Index + Payoff, Strike Capital protection
7. Single Index + Barrier Payoff, Capital protection
8. Single Index + ongoing Coupon, Capital protection
9. Single Index + Payoff, Capital protection
10. Single Index, Capital risk

The structured products are based on the historic price development of the most important American and European indices such as EURO STOXX, DAX, SMI, S&P, and Dow Jones. We us historical end of day closing prices of 94 products with maturities ranging from 2-5 years from Interactive Brokers. Our methodology comprises the daily closing prices of the above mentioned structured products for the regression analysis, the daily closing prices of the indices for the calculation of the fair value, and the daily option closing prices for the calibration of Heston parameters. The payoffs were calculated using the descriptions, resp. formulas given in the respective term sheet of the corresponding product. We use the following abbreviations, applicable to all products: T: Maturity, IC: Invested notional, CP: Capital protection, C_t: Coupon at time t, L_t: Level at time t, B_t: Barrier at time t, r_t: Risk-free rate at time t, $P_i(t)$: Price of underlying i at time t, t_i i-th observation with coupon payment.

3.3 Structured products

3.3.1 Dual Index kick-out, Capital risk

This product offers a pre-determined coupon at a specific observation date in the future, if both indices close at or above a pre-determined level. In that case, the payoff at time τ is calculated as follows:

$$\text{Payoff}(\tau) = \min_{t>\tau}\{C_t * e^{-r_\tau(t-\tau)}\} + IC * e^{-r_\tau*(T-\tau)}, t : P_1(t) > L_1(t) \wedge P_2(t) > L_2(t) \tag{78}$$

Otherwise, the investor bears the following capital risk:

$$\text{Payoff}(\tau) = IC * \min\left\{\frac{P_1(T)}{P_1(t_0)}, \frac{P_2(T)}{P_2(t_0)}\right\} * e^{-r_\tau(T-\tau)} \tag{79}$$

3.3.2 Dual Index + Coupon, Capital risk

If both indices trade during the whole life time T of the product always above the corresponding, pre-determined levels, then the investor obtains the following payoff at time τ:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{T} C_t * e^{-r_\tau*(t-\tau)} + IC * e^{-r_\tau*(T-\tau)} \tag{80}$$

If one of both indices touches or drops below the corresponding level, then the investor bears the following capital risk:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{T} C_t * e^{-r_\tau*(t-\tau)} + IC * \min\left\{\frac{P_1(T)}{P_1(t_0)}, \frac{P_2(T)}{P_2(t_0)}\right\} * e^{-r_\tau*(T-\tau)} \tag{81}$$

3.3.3 Triple Index + Coupon, Capital risk

If all indices trade at a pre-determined observation date at or above a pre-determined level, then the investor gets the following coupon until early redemption:

$$C_t = IC * (N + 1), \tag{82}$$

where N defines the amount of not-paid coupons until this pre-arranged observation date. If all indices trade on a pre-determined observation date T_l at or above a

pre-determined barrier, then an early redemption happens and the product is closed. The payoff at this time τ is then:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{t_n} C_t * e^{-r_\tau * (t-\tau)} + IC * e^{-r_\tau * (T_l - \tau)} \tag{83}$$

Otherwise the investor bears the following capital risk:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{t_n} C_t * e^{-r_\tau * (t-\tau)} + IC * \min\left\{\frac{P_1(T)}{P_1(t_0)}, \frac{P_2(T)}{P_2(t_0)}, \frac{P_3(T)}{P_3(t_0)}\right\} * e^{-r_\tau * (T-\tau)}, \tag{84}$$

where n is the amount of paid out coupons until maturity.

3.3.4 Single Index + Coupon, Capital risk

If the index closes on a pre-determined observation date above a pre-determined level, then the investor gets the following coupon until early redemption of the product:

$$C_t = IC * (N + 1), \tag{85}$$

where N is the amount of non-paid coupons until this pre-determined observation date. If the index closes at a ore-determined observation date T_l above a initial level, then the product expires early and the investor gets back his invested money:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{t_n} C_t * e^{-r_\tau * (t-\tau)} + IC * e^{-r_\tau * (T_l - \tau)} \tag{86}$$

Otherwise, the investor bears the following capital risk:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{t_n} C_t * e^{-r_\tau * (t-\tau)} + IC * \frac{P_1(T)}{P_1(t_0)} * e^{-r_\tau * (T-\tau)} \tag{87}$$

3.3.5 Single Index + maturity Coupon, Capital protection

This product offers a X % capital protection with following payoff:

$$\text{Payoff}(\tau) = (KP + IC * max((P_1(T) - L_T), 0)) * e^{-r_\tau * (T-\tau)} \tag{88}$$

3.3.6 Single Index + Payoff, Strike Capital protection

The investor gets a pre-determined coupon C_T at the end of the maturity if the index hits the pre-determined level of drops below it once. Additionally, the invested capital is protected with 100% as follows:

$$\text{Payoff}(\tau) = C_T * e^{-r_\tau * (T-\tau)} + IC * (1 + max(\frac{P_1(T) - B_T}{P_1(t_0)}, 0)) * e^{-r_\tau * (T-\tau)} \quad (89)$$

3.3.7 Single Index + Barrier Payoff, Capital protection

This 100% capital protected product generates the following payoff if the price of the index trades at or above the strike during the whole lifetime of the product:

$$\text{Payoff}(\tau) = (C_T + IC) * e^{-r_\tau * (T-\tau)} \quad (90)$$

Otherwise the investor obtains the following payoff:

$$\text{Payoff}(\tau) = IC * (max(\frac{P_1(T) - B_T}{P_1(t_0)}, 1)) * e^{-r_\tau * (T-\tau)} \quad (91)$$

3.3.8 Single Index + ongoing Coupon, Capital protection

If the index closes on a pre-determined observation date above a pre-determined level, then the investor gets the following coupon until early redemption of the product:

$$C_t = IC * (N + 1), \quad (92)$$

where N is the amount of non-paid coupons until this pre-determined observation date. If the index closes at a ore-determined observation date T_l above a initial level, then the product expires early and the investor gets back his invested money:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{t_n} C_t * e^{-r_\tau * (t-\tau)} + IC * e^{-r_\tau * (T_l - \tau)} \quad (93)$$

Otherwise, the investor gets the following capital protect payoff:

$$\text{Payoff}(\tau) = \sum_{t=t_1}^{t_n} C_t * e^{-r_\tau * (t-\tau)} + IC * (max(\frac{P_1(T) - B_T}{P_1(t_0)}, 1)) * e^{-r_\tau * (T-\tau)} \quad (94)$$

3.3.9 Single Index + Payoff, Capital protection

With this product, the investor's capital is is 100% protected. Additionaly, the investor obtains the following payoff:

$$\text{Payoff}(\tau) = IC * (max(\frac{P_1(T)}{P_1(t_0))}, 1)) * e^{-r_\tau * (T-\tau)} \tag{95}$$

3.3.10 Single Index, Capital risk

This product offers no capital protection, but offers a participation in the EuroStoxx performance:

$$\text{Payoff}(\tau) = IC * \frac{P_1(T)}{P_1(t_0))} * e^{-r_\tau * (T-\tau)} \tag{96}$$

3.4 Pricing

[Wallmeier and Diethelm(2009)] develop a clever valuation technique based on a multinomial tree and use it to examine how multiple barrier reverse convertibles (MBRCs) are priced in the market. They show that prices exceed model values on average, with greater overpricing when the stocks are less commonly used in MBRCs and for those denominated in low interest rate currencies.

Since some of the products include multiple assets, we use a multivariate geometric Brownian motion in order to price and reproduce the dependency of the underlyings amongst each other:

$$dS_t^i = X_t^i dt + \sigma_i S_t^i dW_t^i, \tag{97}$$

here the Wiener processes are correlated such that $\mathbb{E}(dW_t^i, dW_t^j) = \rho_{i,j}, dt$, where $\rho_{i,i} = 1$. This enables us to calculate a more precise fair value. For calculation of the covariance matrix and μ of the multivariate geometric Brownian motion, we take historic data of the past year. Our total data availability ranges from 2-5 years of historic data, where we use daily closing prices.

For products consisting of only one underlying, S_t we use the Heston model for pricing:

$$dS_t = \mu S_t dt + \sqrt{\nu_t} t S_t dW_t^S, \tag{98}$$

where ν_t, the instantaneous variance, is a CIR process:

$$d\nu_t = \kappa(\theta - \nu_t)\, dt + \xi\sqrt{\nu_t}, dW_t^\nu, \tag{99}$$

and W_t^S, W_t^ν are Wiener processes (i.e., random walks) with correlation ρ, or equivalently, with covariance ρ dt, where μ is the rate of return of the asset, θ is the long run variance, κ is the rate at which ν_t reverts to θ, ξ is the volatility of the volatility and determines the variance of ν_t.

For the calibration of the Heston parameters, we perform Levenberg Marquardt in combination with Genetic algorithms, since this is the best method in order to minimize the root mean square error between the estimated plain vanilla Heston option price and the realized option price of option i on the estimation day:

$$arg\ min_\Omega \sqrt{\frac{\sum_{i=1}^{N}(C_0(i,r,M_i,S,K_i)-C_i)^2}{N}}, \tag{100}$$

where Ω is the set of Heston parameters to be estimated, N the number of options on the estimation day, C_0 is the Heston call function denotes the dollar adjusted call plain vanilla option price, r interest rate, M_i the Maturity of Option i, S the closing price of the Underlying, K_i the strike of option i.

We showed in [Avdiu(2017)] that options with shorter maturities, at-the-money and out-of-the money options yield the most accurate fair values. Thus, we take the best 23 options of the last 5 days in order to obtain the best calibration results.

We simulate for each underlying one 500,000 random path. Then we apply the payoffs of all ten products to these simulated paths. We then take the average of all outcomes to calculate the fair value.

3.5 Driver analysis

In order to analyze the explanatory power of volatility and liquidity on the markup, we regress each structured product markup on the corresponding one-week volatility and liquidity. This allows us to extract the drivers of markup size. Thus, we use volatility and liquidity as a proxy for market trust and regress $Markup_{i,t}$ for structured product i at time t on the volatility $\sigma_{i,t}$, resp. on the liquidity $\lambda_{i,t}$, considering time lags.

3.5.1 Market liquidity estimation

For calculation of the market liquidity we use a measure which generates an estimated traded volume per time unit, as described in [Avdiu(2017)]. The Heston model assumes a stochastic volatility development of the bid (i) and ask (j) prices $(S_{i,j})$ with parameters $\mu_{i,j}$(bid/ask price drift), $V(t)_{i,j}$(bid/ask price variance), κ

(rate of mean reversion), $\omega_{i,j}$(long run variance), $\sigma_{i,j}$(volatility of variance) and $W_{1,2}^{i,j}$ (Standard Brownian movements). Thus we take for the bid-ask dynamics:

$$\frac{dS(t)_{ij}}{S(t)_{ij}} = \mu_{ij}dt + \sqrt{V(t)_{ij}}dW_1^{ij}, \tag{101}$$

$$dV_{ij} = \kappa_{ij}(\omega_{ij} - \sigma_{ij})dt + \sigma(t)_{ij}\sqrt{V(t)_{ij}}dW_2^{ij}, \tag{102}$$

where W_1 and W_2 are correlated by $dW_1 \cdot dW_2 = \rho dt$ due to the leverage effect between asset price and instantaneous volatility. For estimation of the traded bid-ask volumes we use inverse transform sampling which we apply to the historic data:

$$P(Q_{i/j}(t) = x_{i/j}^k) = \sum_{t=1}^{k} p_{i/j}^t - \sum_{t=1}^{k-1} p_{i/j}^t = p_{i/j}^k \tag{103}$$

The resulting compound volume process $Y(t)$ at time t characterizes the volume generating process induced by trading.

$$Y(t) = \begin{cases} min\{Q_i(t), Q_j(t)\} & \text{if } S_i(t) \geq S_j(t) \\ 0 & o.w. \end{cases}. \tag{104}$$

By matching bid and ask prices and taking the average of each possible generated volume we get the average traded volume (liquidity) over a certain time period n:

$$\lambda = \frac{\sum_{t=1}^{n} Y(t)}{n} \tag{105}$$

3.5.2 Multivariate driver analysis

For structured products with multiple underlyings we conduct a multivariate regression for $Markup_{i,t}$ for product i at time t, on the volatility $\sigma_{j,t}$ of underlying j:

$$Markup_{i,t} = \alpha + \sum_{j=1}^{N} \gamma_j \sigma_{j,t} + \epsilon_{i,t}. \tag{106}$$

Then we run a regression on liquidity $\lambda_{j,t}$ of product i of underlying j:

$$Markup_{i,t} = \alpha + \sum_{j=1}^{N} \gamma_j \lambda_{j,t} + \epsilon_{i,t}. \tag{107}$$

3.5.3 Single driver analysis

For single products with just one underlying we conduct a standard regression based on the volatility $\sigma_{i,t}$ of product i:

$$Markup_{i,t} = \alpha + \gamma_p \sigma_{i,t} + \epsilon_{i,t}. \tag{108}$$

To check for liquidity as driver of the dynamics in the markup of single underlying structured products, we also need to regress $Markup_{i,t}$ for structured product i at time t on the liquidity $\lambda_{i,t}$ over time:

$$Markup_{i,t} = \alpha + \gamma_p \lambda_{i,t} + \epsilon_{i,t}. \tag{109}$$

3.6 Results

We calculate the results for all ten structured products, but just present here graphically the first product as an example.

3.6.1 Example 1: Dual Index kick-out, Capital risk

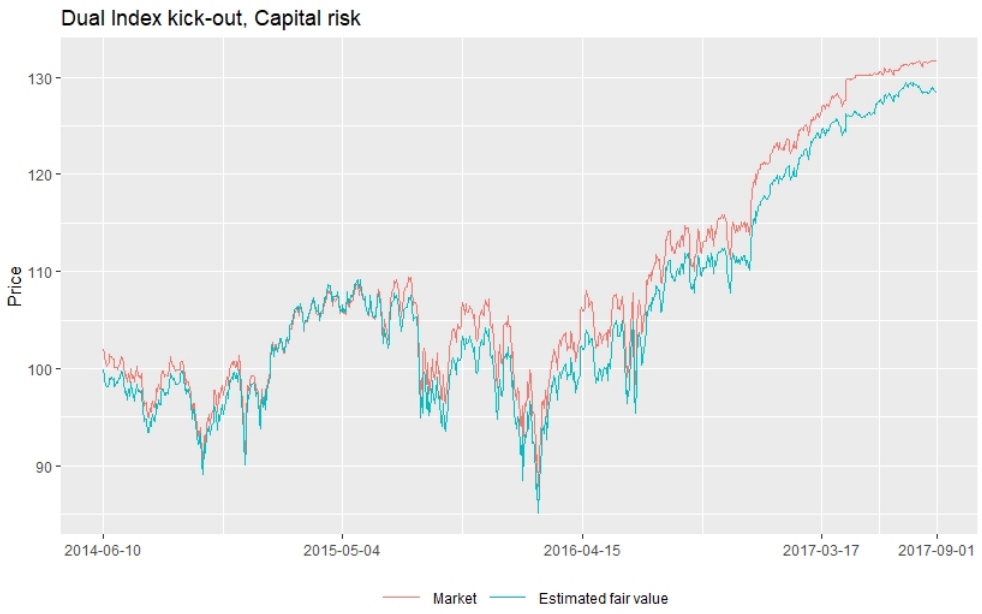

Figure 21: Calculated fair value vs.historic traded price.

For the Dual Index kick-out, Capital risk, we calculate over the past 4 years its historic fair value, using a calibrated Heston model, and compare it to the historic traded prices of the structured product. We can see that the traded market prices exhibit a significant markup in price which reflects the implicit hedging costs for this product, as well as the banks' premium.

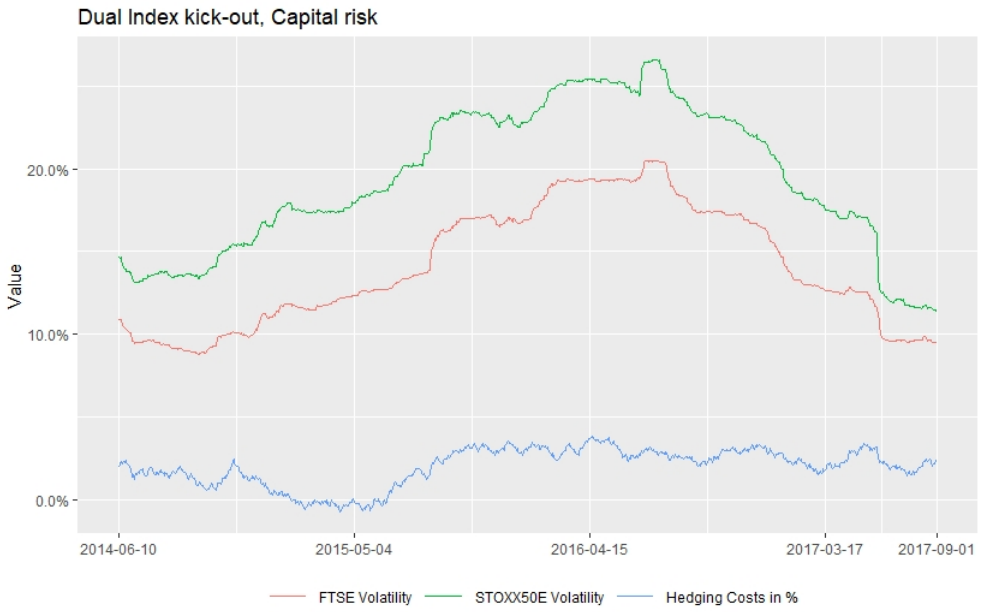

Figure 22: Volatility of FTSE and ESTX50 future, with corresponding implicit hedging costs.

When looking at the volatility of the underlyings of the Dual Index kick-out, Capital risk, we can first of all see a positive correlation between both indices. When we plot the corresponding hedging costs over time, we can also see an increase in hedging costs with an increase in volatility.

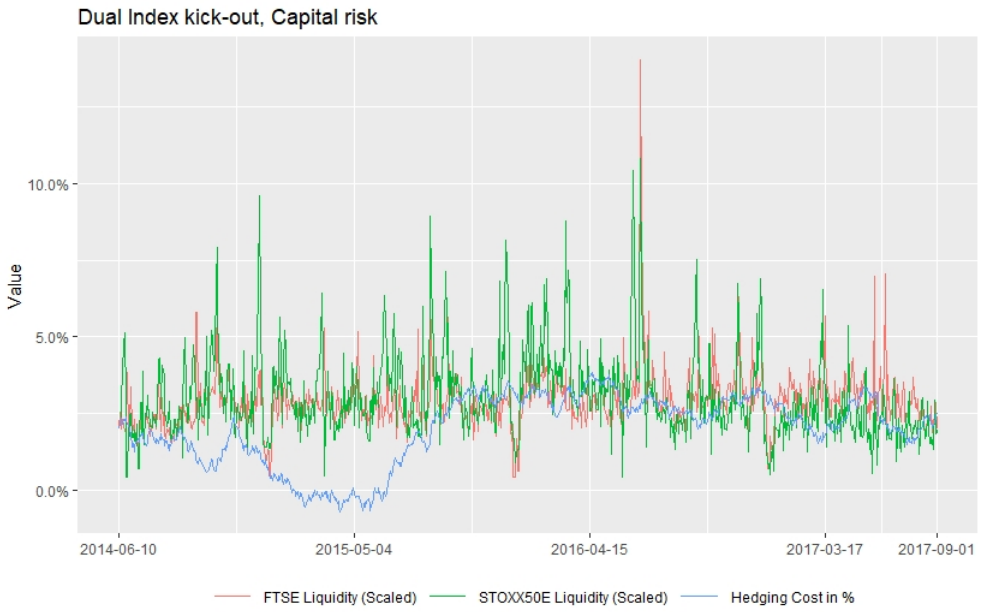

Figure 23: Liquidity of FTSE and ESTX50 future, with corresponding implicit hedging costs.

When looking at the market liquidity of the underlyings of the Dual Index kick-out, Capital risk, we don't see any correlation or direct relationship between the liquidity of both indices and the corresponding hedging costs.

3.6.2 Hedging and Liquidity Costs

The 10 structured products were replicated on the basis of their payoffs and priced with multivariate geometric Brownian motion or the optimal Heston Model (fair value).

The difference between Fair Value and historical prices is defined as Hedging and Liquidity Costs (Markup).

Table 20: Hedging and Liquidity Costs

Product	Min	Max	Average
Dual Index kick-out, Capital risk	-0.7%	3.8%	2.0%
Dual Index + Coupon, Capital risk	0.1%	1.4%	0.9%
Trippel Index + Coupon, Capital risk	1.3%	3.1%	2.9%
Single Index + Coupon, Capital risk	-0.2%	1.6%	1.4%
Single Index + maturity Coupon, Capital protection	-0.3%	1.8%	1.3%
Single Index + Payoff, Strike Capital protection	-0.7%	3.5%	2.1%
Single Index + Barrier Payoff, Capital protection	0.3%	2.9%	1.4%
Single Index + ongoing Payoff, Capital protection	0.5%	2.9%	1.9%
Single Index + Payoff, Capital protection	0.6%	4.1%	2.2%
Single Index, Capital risk	-0.1%	0.5%	0.2%

Table 20 shows the range and average markup on each structured product. We can see that the average markup on a structured product ranges from 0.9% to 2.9%.

3.7 Conclusion

We replicate the payoff of ten different structured products and compare their calculated fair values with the real traded prices in order to determine the markup incurred in the products. We find significant markups, ranging from 0.9%-2.9% on the fair value, which serves as an indication for implicit hedging costs associated for the corresponding product. The markup also includes any premium earned by the issuing bank, but it is not possible to infer the share how much of that markup can be assigned towards he hedging costs and how much towards the bank premium.

Overall we find that the better the payoff of the product for the investor the higher the hedging costs for the bank. This means that clients face a higher capital protection, resp. lower barriers that trigger a positive payoff.

We are the first to use a calibrated Heston model to replicate the payoffs of single asset structured products, which yields better results than pricing methods using standard Brownian motions over a long period of time . The replication is done by using cluster calculation, a new technology using parallel computing, which was not possible in the past due to the unavailability of the technology.

Based on these results it is possible for us to identify possible drivers for this markup, especially if these drivers explain the change in markup over time. We test two possible drivers: Volatility and market liquidity of the underlying. We find significant results for volatility for 8 of 10 products, while we only identify liquidity

as a driver for one product, where the payoff of this product does not require any hedging.

Our results indicate that most of the markup can be assigned to hedging costs, while hedging costs due to market liquidity can't be extracted in complex payoffs, as the underlying's volatility seems to dominate the cost of hedging in complex structures. In the presence of hedging, liquidity costs are embedded in the underlying's hedging costs.

We find the main driver of the markup of complex structured products to be volatility. The liquidity costs can only be extracted from non-complex structures as we find it to be the single driver of product markups. Only in absence of hedging necessity, market liquidity seems to drive the dynamics of the markup changes.

We can conclude that products with hedging requirements are exposed to volatility, as it drives the hedging costs, while the markup of products with no hedging requirements can be explained by market liquidity.

3.8 Appendix

Table 21: Volatility

Product Group	Nr. of products	Correlation	p-value	R-squared
Dual Index kick-out, Capital risk	15	0.4-0.78	0.02-0.06	0.42-0.68
Dual Index + Coupon, Capital risk	12	0.42-0.62	0.03-0.04	0.55-0.68
Trippel Index + Coupon, Capital risk	7	0.55-0.84	0.06-0.09	0.53 - 0.78
Single Index + Coupon, Capital risk	13	0.42-0.65	0.008-0.02	0.55 - 0.88
Single Index + maturity Coupon, Capital protection	7	0.52-0.68	0.03-0.06	0.45 - 0.38
Single Index + Payoff, Strike Capital protection	5	0.61-0.89	0.007-0.02	0.61 - 0.91
Single Index + Barrier Payoff, Capital protection	4	0.55-0.81	0.01-0.03	0.48 - 0.62
Single Index + ongoing Coupon, Capital protection	9	0.61-0.78	0.01-0.04	0.53 - 0.78
Single Index + Payoff, Capital protection	13	0.62-0.9	0.008-0.02	0.57 - 0.81
Single Index, Capital risk	9	0.21-0.32	0.08-0.2	0.15 - 0.26

References

Avdiu K., Sept. 2017, Liquidity estimation in a high frequency setup, Unpublished working paper.

Bergstresser D., 2008, The retail market for structured notes: issuance patterns and performance, 1995â2008. Unpublished working paper, Harvard Business School.

Burth Stefan, Kraus Thomas and Wohlwend Hanspeter, 2016, The Pricing of Structured Products in the Swiss Market, The Journal of Derivatives Winter 2001, 9 (2) 30-40; DOI: https://doi.org/10.3905/jod.2001.319173

Henderson Brian J. and Pearson Neil D., 2011, The dark side of financial innovation: A case study of the pricing of a retail financial product, Journal of Financial Economics, Volume 100, Issue 2, pp. 227-247.

Jarrow R. and O'Hara M., 1989, Primes and scores: an essay on market imperfections, Journal of Finance, 44, pp. 1263-1287.

Rogalski R. and Seward J., 1991, Corporate issues of foreign currency exchange warrants Journal of Financial Economics, 30, pp. 347-366.

Stoimenov Pavel A. and Wilkens Sascha, 2005, Are structured products 'fairly' priced? An analysis of the German market for equity-linked instruments, Journal of Banking & Finance, Volume 29, Issue 12, December 2005, pp. 2971-2993.

Sascha Wilkens, Carsten Erner, and Klaus Roeder, 2003, The Pricing of Structured Products - An Empirical Investigation of the German Market, The Journal of Derivatives.

Wallmeier Martin and Diethelm Martin, 2009, Market Pricing of Exotic Structured Products: The Case of Multi-Asset Barrier Reverse Convertibles in Switzerland, The Journal of Derivatives Winter, 17 (2) 59-72; DOI: https://doi.org/10.3905/JOD.2009.17.2.059

4 Minimizing the Index Tracking Error using Optimization Methods

Kujtim Avdiu[1] and Stephan Unger[2*]

[1] Nationalbank Austria, Otto-Wagner-Platz 3,
1090 Vienna, Austria, kujtim.avdiu@oenb.at, Mobile: +43 676 9722535
[2] Saint Anselm College, Department of Economics & Business, Saint Anselm Drive 100, Manchester, NH, USA, email: sunger@anselm.edu,
Phone: +1 603 641 7312, Mobile: +1 646 991 8501
*Corresponding author

Abstract

We propose a new approach for index tracking error minimization by using the correlation factor in the target function. We analyze different optimization techniques for minimizing the tracking error of equity and fixed income ETFs. We test the DAX and a bond ETF empirically and also simulate a bond ETF using the CIR model for the modeling of the short-rate. We show that by making use of these techniques an approximation of the stock as well as bond index performances by an ETF is possible. We find that the Simplex method yields the smallest tracking error for replicating the stock index performance, resp. the Levenberg-Marquardt method for replicating the bond index performances.

Keywords: ETF tracking, optimization techniques, Levenberg-Marquardt, Nelder-Mead Simplex.

JEL-Codes: C61, G11, C52.

Author contribution
Based on the paper of Jeurissen and van den Berg [3], S.U. proposed the idea to investigate the tracking error for bond and equity indexes. K.A. added to this idea the application of optimization methods. K.A. developed the model and proposed to use the correlation as target function. S.U. provided and analyzed the data, while K.A. performed the calculations and programming tasks. S.U. & K.A. interpreted the results and summarized the conclusion together.

Submitted: Annals of Operations-Research

4.1 Introduction

Most optimization techniques are dealing with minimization of some generated data according to a target function, which is in most cases comprised of realized observations. The minimization of the distance between forecasted, or estimated values, and realized values needs a proper framework for which these techniques offer robust solutions. A very important applied optimization problem is the minimization of the tracking error of ETFs.

Tracking error is a real problematic issue for ETF investors. ETFs are designed with the objective of replicating the performance of specific indices. The primary goal of portfolio managers is to minimize the tracking error, which is the performance difference between an ETF's net asset value (NAV) and its underlying index. Therefore, tracking error can indicate to us how well managers have met their objectives. Factors such as fees and expenses, creation and redemption costs, portfolio optimization, index changes and dividend reinvestment policy may cause an ETF's performance to deviate from its underlying index.[1]

A quite a novel instrument for tracking the fixed income market are fixed income ETFs. According to a definition of Fidelity: "Fixed income exchange-traded funds (ETFs) are baskets of fixed income investments that trade as a single unit throughout the day. That means, by creation of a fixed income ETF, intra-day visibility arises. This encompasses not only intra-day price movements but also cost transparency. Therefore FI-ETFs belong to an emerging product family which investors are seeking with increasing interest.

Generally there exist two types of ETFs managers: An active manager tries to select stocks from the index that will outperform others in the future and thus beat the performance of the index. Passive managers, on the other hand, do not believe in beating the market permanently and therefore try to replicate the market 100 %. [13]

For passive management of ETFs there are 2 possible ways: Full replication and partial replication. Full replication, selects all stocks that make up an index in their respective market weighting. In contrast, partially replicated ETFs try to approximate the underyling's index performance as good as possible by investing in a minimum amount of underlyings that map the index best. This gives the manager more flexibility and offers him a reduced cost structure which allows him to handle the asset allocation in a way that the portfolio might not be as vulnerable to declines in the underlying index as the index itself. On the downside, the tracking error of such passively managed indices is larger than the passively managed ones, which makes the minimization of the tracking error an important challenge.

Therefore it makes sense to investigate and rate different optimization techniques which could be applied in order to minimize the tracking error. For reducing the

tracking error several optimization methods can be applied, such as Genetic algorithms (GA), the Particle Swarm Optimization technique (PSO), the Levenberg-Marquardt method (LM) and the Nelder-Mead Simplex method (NMS).

Jeurissen and van den Berg [3] found that Index tracking can be significantly improved by using hybrid genetic algorithms in order to reduce the tracking error. This method showed to outperform a randomly selected portfolio which was used to track the performance of the AEX-index. For further ETF-tracking literature we also refer to [3].

Our approach builds up on this idea by applying four different optimization techniques to a bond index tracking. Beside the information about quality of these optimization techniques, we propose a new approach for index tracking error minimization by using the correlation as target function. A tracking error minimization means pushing $\rho_{(Index,ETF)} \to 1$ equivalently.

Moreover, we are the first to apply this methodology to a real traded stock and bond ETF as well as to a simulated bond ETF and to compare their tracking errors with the corresponding traded ETF indices.

This study shows uniquely which calibration technique should be used in order to minimize the ETF tracking error and compares the tracking performance between a real traded stock ETF and its index, as well as a simulated bond ETF and the corresponding bond ETF using real market data.

Since we use a randomly generated benchmark as well as realized market data the results obtained can be applied to any upcoming index tracking problem. For the stock ETF index we use the historical end-of-day traded prices of the 30 DAX stocks to replicate the ETF index using a combination of 12, 15, 18 and 21 stocks to test the tracking error minimization capabilities of the above mentioned four optimization methods. For the simulated fixed-income ETF we virtually issue a realized historical and a simulated bond index for which we use the CIR-model in order to simulate the short-rate for pricing 10 bonds with 10 years maturity, but different credit spreads, to test combinations of 4, 5, 6 and 7 bonds. For the real traded bond ETF we take 10 randomly chosen bonds due to unavailability of historically real traded bond ETF data on a sub level. Therefore we construct an own bond index by choosing random weights.

We want to track the ETF as close as possible by a one-time asset allocation which is done in advance. The simulation and evaluation is performed 10.000 times. The average resulting tracking errors are investigated for each optimization technique and evaluated for effectiveness and robustness for any asset allocation.

All the optimization techniques are performed in order to minimize the tracking error between the optimal allocated stock ETF and real DAX ETF as well as the optimal bond ETF and the virtually constructed bond index. We study the effec-

tiveness and robustness of these four optimization techniques for performing future ETF allocations.

This paper is structured as follows: In chapter (4.2) we simulate 10 bonds with maturity 10 years but different credit spreads by using a calibrated CIR model and yield curve EURIBOR data from Bloomberg in order to estimate the parameters of the CIR model. For the real traded bonds and assets we use historic prices from Bloomberg. In chapter 4.3 construct a bond index and project a possible development of this bond index within the next 5 years. Moreover we construct various ETFs allocated with a different number of bonds and stocks, and perform four different optimization methods in order to find out which method generates the best tracking error. The optimization problems are mentioned in chapter 4.4 for each optimization method. Chapter 4.5 presents the results obtained and chapter 6 concludes.

4.2 Bond index pricing

The Cox-Ingersoll-Ross (CIR) model assumes that the short-rate, r_t, satisfies

$$dr_t = \alpha(\mu - r_t)dt + \sigma\sqrt{r_t}dW_t,\ \alpha, \mu, \sigma > 0, r \geq 0, \tag{110}$$

where W_t is a standard Brownian motion, α is the speed of adjustment, μ is the long term average rate (the mean-reverting level) and $\sigma\sqrt{r_t}$ is the implied volatility. Within the CIR model the short rate is guaranteed to remain non-negative. We compute the zero-coupon bond prices by

$$P_t^\tau = E_t^*[exp(-\int_t^{t+\tau} r_u du)], \tag{111}$$

where E^* denotes expectation under the risk-neutral probability measure Q^*. The pricing relation shows that any yield-curve model consists of two ingredients:

- the change of measure from Q to Q^* and
- the dynamics of the short rate r under Q^*.

Assuming that the Local Expectation Hypothesis holds, an advantage of the LEH is that there is no need to know how to change the probability measure in step (i). Another advantage is that we have some intuition about the parameters that determine the dynamics of the short rate under the data-generating measure, whereas we do not have such intuition about the parameters under the risk-neutral measure. The LEH is therefore a useful starting point.[5]

The LEH states that:

$$E(\frac{dP}{P}) = r_t + \pi r_t \frac{P_r}{P}, \tag{112}$$

where $P(t,T)$ is the price at time t of a zero-coupon bond with maturity T and π is the risk premium. It is possible to prove that

$$P(t,T) = F(t,T)e^{G(t,T)r_t}, \tag{113}$$

where

$$F(t,T) = (\frac{\phi_1 e^{\phi_2(T-t)}}{\phi_2(e^{\phi_1(T-t)}-1)+\phi_1})^{\phi_3}, \tag{114}$$

$$G(t,T) = (\frac{e^{\phi_1(T-t)}-1}{\phi_2(e^{\phi_1(T-t)}-1)+\phi_1}) \tag{115}$$

and

$$\phi_1 = \sqrt{(\alpha+\pi)^2+2\sigma^2}, \tag{116}$$

$$\phi_2 = \frac{\alpha+\pi+\phi_1}{2}, \tag{117}$$

$$\phi_3 = \frac{2\alpha\mu}{\sigma^2}. \tag{118}$$

CIR's basic point is that when interest rates are random, different term premia cannot simultaneously equal zero because of Jensen's Inequality. Therefore, different versions of the pure expectations theory are inconsistent with one another. But Campbell [6] argues that "it turns out that different versions of the expectations theory, as opposed to the pure expectations theory, are not necessarily incompatible with each other or with arbitrage pricing equilibrium". He shows that even if "Jensen's Inequality places a wedge between different concepts of risk premia in the term structure, this does not prevent different premia from being constant through time at different values".

For the yield-curve fitting we use the simplex method since due to [7] "it is one of the best curve fitting methods, having several advantages. First, it is simple to use. Compared to the well-known least-squares method, the simplex method does not need the derivative function, nor does the orthogonality condition apply. Second, it is easy to adopt any function in the computer program-only the function evaluation needs to be changed. And since the response surface is a paraboloid, there are no local minima giving false solutions. However, the simplex calculation is a little slower than the analytic least-squares calculation."

We run 20.000 iterations to get following parameter estimates for the CIR model per Dec 8, 2014: $r_0 = 0.0001890656$ $\phi_1 = 0.165459$ $\phi_2 = 0.1$ $\phi_3 = 0.342016$ $\pi = -0.649334$ $\alpha = 0.365147$ $\mu = 0.0424216$ $\sigma^2 = 0.00230475$. Bootstrapping using these parameters we get following corresponding estimated yield-curve in Figure 24.

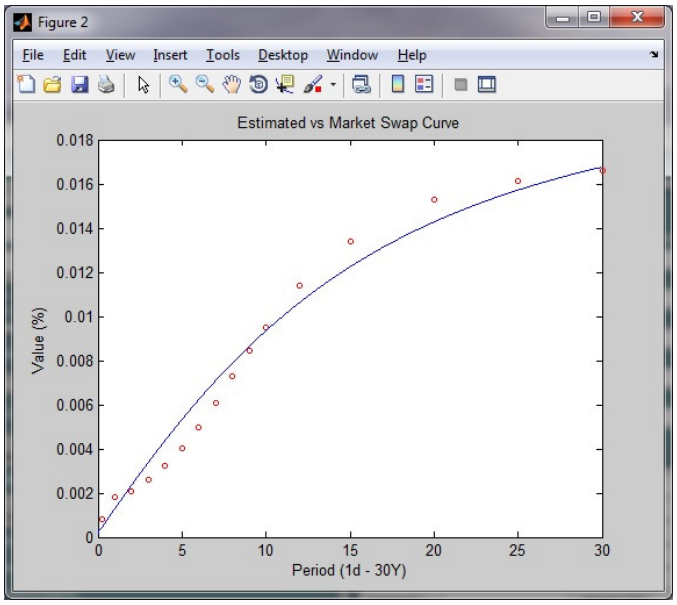

Figure 24: Fitted Euro yield-curve

4.3 Index and ETF construction

4.3.1 Benchmark projection

Fixed-income benchmark

After parameter estimation of CIR model and determination of the yield-curve we are now able to build the simulated price of a bond index at time t. Our example focuses on an index consisting of 10 bonds, each of maturity 10 years for 1250 trading days which is calculated by

$$Index_t = Index_{t-1} \sum_{n=1}^{10} w_n \frac{B_n(t)}{B_n(t-1)}, \tag{119}$$

where w_n is the n-th weight of the corresponding bond and $B_n(t)$ is the n-th bond price at time t with maturity 10 years and $Index_1 = 100$. In order to generate the simulated bond price for these 5 years we build 10 paths, one for each bond, where each path represents a possible bond price development until maturity. For simulation we assume a randomly chosen vector of credit spreads from 2% to 3%.

Table 22 outlines the composition of the bond index using real market data.

Table 22: Bond index composition

ISIN	Name	Emmitent	Maturity
XS0122710188	Unicr.Bk Aus. 01/31Mtnflr	UNICREDIT BK AUS	23.01.2031
DE000A0XYJ16	Kreditanst.f.Wiederaufbau	KFW	19.05.2019
XS0160028014	Abruzzo, Reg. 02/36Flrmtn	REGION OF ABRUZZO	06.11.2036
XS0162062888	Puglia, Reg. 03/23Flrmtn	REGION OF PUGLIA	05.02.2023
XS0286746432	Sns Bank Nv 07/17 Flr Mtn	SNS BANK NEDERLAND	12.02.2017
XS0282583722	Morgan Stanley 07/17 Flr	MORGAN STANLEY	15.01.2017
XS0284728465	Goldm.S.Grp 07/17 Flr Mtn	GOLDMAN SACHS	29.01.2017
ES0214958102	OS CAIXANOVA-A VBLE	ABANCA CORP BAN.	20.02.2017
ES0000095812	Gen.Catalunya	GENERALITAT DE CATAL.	12.06.2018
XS0300196879	Intesa San.07/17 Flr Mtn	WMEM850605	17.05.2017

For both cases market and simulated data we use the same randomly chosen weight vector, $w = (0.2, 0.15, 0.1, 0.07, 0.09, 0.12, 0.15, 0.03, 0.02, 0.07)$.

The resulting projected benchmark for our simulated test environment can be seen in Figure 25.

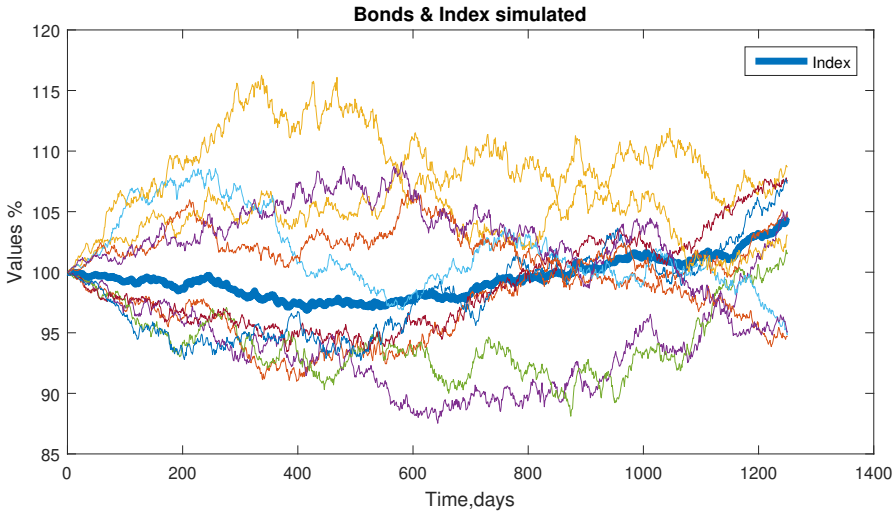

Figure 25: 1250 simulated trading days projected index

The resulting projected benchmark for our market data test environment can be seen in Figure 26.

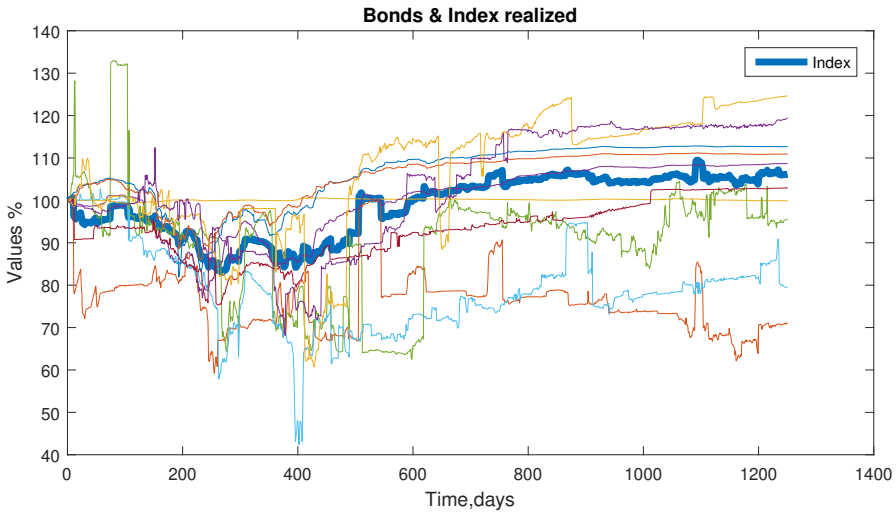

Figure 26: 1250 historical trading days projected index

Depending on their financial situation, banks and corporates are able to adjust the yields of their newly issued bonds due to changes in default or risk premiums. This change has an impact on the price development of bonds trading in the secondary market, as the jumps in Figure 26 indicate. Both factors were neglected in the simulation of the bond prices.

Equity benchmark

In order to test our methodologies on real data we apply the four optimization techniques to daily DAX ETF index data. The goal is to track the performance of the DAX ETF with less stocks than the actual 30 DAX stocks to save transaction costs during the readjustment processes. Since the combination of all stocks for different weights applying all four optimization methods exceeds even regular cluster calculation capacities, we cap our maximum number of combinations at 1,200 concerning the stocks. The choice of the combination of stocks is done with same probability. The reason for this cap arises from the fact that no significant contribution to minimize the tracking error can be achieved after 1,200 combinatorial trials. This means that the identification of the stocks which track the DAX ETF index best is done through the application of our four optimization methods, i.e. by optimizing the weights which minimize the tracking error, capped at 1,200 trials.

Table 23 outlines the composition of the DAX ETF index.

Table 23: DAX ETF index composition

1. Linde plc	16. Bayer Aktiengesellschaft
2. Covestro AG	17. Fresenius SE and Co. KGaA
3. SAP SE	18. Deutsche Telekom AG
4. Deutsche Borse AG	19. Adidas AG
5. Siemens Aktiengesellschaft	20. Deutsche Post AG
6. E.ON SE	21. Allianz SE
7. MERCK Kommanditgesellschaft auf Aktien	22. Bayerische Motoren Werke AG
8. BASF SE	23. HeidelbergCement AG
9. Volkswagen AG	24. Deutsche Lufthansa AG
10. Henkel AG and Co. KGaA	25. Daimler AG
11. Deutsche Bank Aktiengesellschaft	26. Beiersdorf Aktiengesellschaft
12. Fresenius Medical Care AG and Co. KGaA	27. Infineon Technologies AG
13. RWE Aktiengesellschaft	28. Munchener Ruckversicherungs-Gesellschaft AG
14. Vonovia SE	29. Continental Aktiengesellschaft
15. Thyssenkrupp AG	30. Wirecard AG

The resulting projected benchmark DAX test environment can be seen in Figure 27.

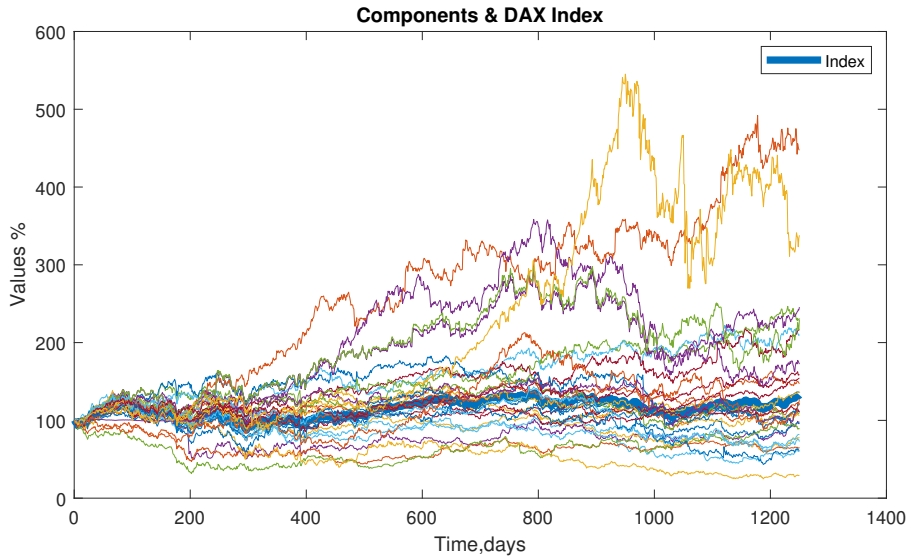

Figure 27: 1250 historical trading days DAX index

4.3.2 ETF construction

The ETF is constructed by allocating different numbers and combination of bonds with different weights. In order to find the optimal allocation the four optimization methods are applied for constructing the ETF. The methods are mentioned in

chapter 4. We test for an interval of 4-7 bonds and 12-21 stocks in step sizes of 3. For each allocation we run the four different optimization methods and evaluate it's tracking error.

4.3.3 ETF tracking error calculation

To calculate Tracking Error, we take the projected daily NAV returns of the bond index and its underlying ETF portfolio allocation for the next 250 trading days and arrange them into matrixes X and Y as follows:

$$X = \begin{bmatrix} R_{ETF_i,1001} \\ R_{ETF_i,1002} \\ R_{ETF_i,1003} \\ . \\ . \\ R_{ETF_i,1250} \end{bmatrix} \quad Y = \begin{bmatrix} R_{Index,1001} \\ R_{Index,1002} \\ R_{Index,1003} \\ . \\ . \\ R_{Index,1250}, \end{bmatrix}$$

where $R_{ETF,t}$ is the return on the ETF-portfolio at time t, containing different bonds and $R_{Index,t}$ is the return on the NAV of the projected index benchmark at time t. Tracking Error is the standard deviation of the active returns (difference between NAV returns and ETF-portfolio returns) calculated as:

$$TrackingError = \sqrt{Var(X - Y)} * \sqrt{250}, \tag{120}$$

The annual multiplier allows an easier easier comparison with other statistical calculations. [8] After 1.000 trading days we calculate the optimal bond weights for tracking the index with a minimum tracking error and apply these weights to the next year, meaning we keep the weights constant for the last 250 trading days for performing an out-of-sample test.

4.4 ETF tracking error optimization

We now have projected an ETF price development which serves as a benchmark for our asset allocation. Our goal is to maximize the correlation between the return of the index price arising from our asset allocation and the realized ETF price:

$$arg\ max_{\Omega}(\rho_{index,ETF}), \tag{121}$$

where Ω is the set of ETF weights to be estimated. Since we want to replicate the bond index by using a new approach we use the correlation between the index and the ETF:

$$\rho_{index,ETF} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{122}$$

where x_i is the i-th return of the projected bond index and y_i is the i-th return of the tracking ETF. We next define for each optimization method the target function compare their accuracy regarding the minimization of the tracking error over 5 years for different bond allocations, combining 4 to 7 bonds, resp. 12-21 stocks.

4.4.1 ETF tracking calibration with Genetic algorithms

Genetic algorithms (GA) operate by maintaining and modifying the characteristics of a set of trial solutions (individuals) over a number of iterations (or generations). Each individual solution is represented by a binary string (or chromosome in biological analogies) in the genetic algorithm. The optimization process is designed to produce, in successive iterations, an increasing number of individuals with desirable characteristics. The process is probabilistic but not completely random. The rules of genetic algorithms have the power of retaining certain desirable characteristics that would otherwise be lost with a completely random searching method. A genetic algorithm consists of the following major components: encoding method, reproduction, crossover, mutation and fitness scale.[9] The target function for our optimization problem using GA is defined by

$$\begin{aligned} f(x) &= -\rho_{(idx,ETF)}, \\ with: &\sum_{i=1}^{n} w_i = 1. \end{aligned} \tag{123}$$

4.4.2 ETF tracking calibration with Particle Swarm Optimization

In Particle Swarm Optimization (PSO, Eberhart and Kennedy, 1995), we have again a population that comprises n solutions, stored in real-valued vectors. In every generation, a solution is updated by adding another vector called velocity v_i. We may think of a solution as a position in the search space, and of velocity as a direction into which the solution is moved. Velocity changes over the course of the optimization, the magnitude of change is the sum of two components: The direction towards the best solution found so far by the particular solution, P_{best_i} , and the direction towards the best solution of the whole population, $P_{best_{gbest}}$. These two directions are perturbed via multiplication with a uniform random variable ξ and constants $c_{(\ddot{\imath}'\,\cdot)}$, and summed. The vector obtained is added to the previous v_i, the resulting updated velocity is added to the respective solution. In some implementations, the velocities are reduced in every generation by setting the parameter Δ, called inertia, to a value smaller than unity [2]. The target function for our optimization problem using PSO is defined by

$$\begin{aligned} f(x) &= -\rho_{(idx,ETF)}, \\ with: &\sum_{i=1}^{n} w_i = 1. \end{aligned} \tag{124}$$

4.4.3 ETF tracking calibration with Levenberg-Marquardt

Levenberg-Marquardt (LM) is an iterative technique that locates the minimum of a function that is expressed as the sum of squares of nonlinear functions. Its principal application is to optimize the parameters $f(x) = \{\rho_1, \rho_2, ...\rho_{10}\}$ of the ETF-tracking portfolio $f(y, x)$, where y is a given variable, so it minimizes the squared distance $\epsilon^T \epsilon$, where $\epsilon = x - \hat{x}$. The Levenberg-Marquardt method performs an interpolation between the Gauss and the steepest gradient descent methods based upon the maximum neighborhood in which the truncated Taylor series gives an adequate representation of the nonlinear model. [10]

In the algorithm a positive constant (damping) is added to the diagonal of of $A^T A$ in order to control the convergence of the method and provide an effective way to avoid the singularity of the system. In the former case damping will determine the rapidness of convergence, with large damping producing slow convergence and vice versa. In the latter case the presence of damping will artificially increase the eigenvalues improving the ill-conditioning of matrix $A^T A$. In the method the step to converge from an initial guess to a final solution is represented by

$$\Delta B = (A^T A + \epsilon^2 I)^{-1} A^T \Delta T, \tag{125}$$

where B= Parameters to find, A= Jacobian matrix, ϵ^2=Damping, T=Data. [11]

We minimize correlation between the index and ETF price using LM with following target function:

$$\begin{aligned} f(x) = min_\Omega[\rho^2_{(idx,ETF)} + 1], \\ with: \sum_{i=1}^{n} w_i = 1, \end{aligned} \tag{126}$$

4.4.4 ETF tracking calibration with Nelder-Mead Simplex

The Nelder-Mead Simplex method (NMS) or Downhill Simplex method is a direct search method which is independent of the existence of derivatives. It solves the optimization problem given a real-valued objective function by starting from a simplex S_0. For each iteration step indexed by k the algorithm identifies a vertex v^k_{max} determined by

$$v^k_{max} = argmax_{x \in \{v^k_0, ... v^k_n\} f(x)}, \tag{127}$$

which is the vertex where the function takes its largest value. The vertex v^k_{max} of the simplex S_k is then replaced by a new point $\hat{v}$ such that $f(\hat{v}) < f(v^k_{max})$. The new simplex is called S_{k+1}. To construct the new vertex $\hat{v}$ of this simplex the method uses three operations called reflection, expansion and contraction.[12]

The target function for our optimization problem using NMS is defined by

$$\begin{aligned} f(x) &= -\rho_{(idx,ETF)}, \\ with: &\sum_{i=1}^{n} w_i = 1. \end{aligned} \tag{128}$$

4.4.5 ETF tracking calibration with Levenberg-Marquardt combined with genetic algorithms

Due to the large amount of data and complexity of the tracking error formula, the calculation run times for minimization of the tracking errors are very high. For this reason, we have also chosen weak termination conditions (Maximum function evaluation, maximum iterations, termination tolerance on the function value). The calibration results are very satisfactory for all optimization methods.

Nevertheless, we also apply a combination of Levenberg Marquardt and Genetic Algorithms by using the optimal parameters of LM as start parameters for GA in order to enhance the minimization of the tracking error. (see results below)

4.5 Results

Using the four optimization methods, we do several asset allocations of 4 to 7 bonds for which for we try to replicate the index as good as possible. We split our projection into two parts: The in-sample test encompasses the first four years (1.000 trading days) whereas the out-of-sample test is performed over the fifth year (250 trading days). We calculate the tracking error for each optimization method and each asset allocation for simulated and market data over the in-sample period as well as over the out-of-sample testing period. It is obvious that for a larger number of bonds used the tracking error decreases. But the interesting point for an ETF-manager is to replicate the index using a minimum number of bonds. Depending on the costs of buying an additional bond compared to the costs of a higher tracking error, the ETF-manager may choose his own allocation, especially with respect to hedging purposes.

Termination tolerance on the function value is 0.001. If the termination condition was not reached, we use the additional optimization parameters from table 24.

Table 24: Termination conditions

Maximum function evaluation	1,000
Maximum iterations	20,000
Termination tolerance on the function value	0.001

Comparing each method by averaging the tracking errors of all asset allocation, we get following tracking errors (TE) for each optimization method in-sample (Table 25):

Table 25: Tracking error in-sample

Method	Simulated Avg. TE	Market Avg. TE	DAX Avg. TE
LM	0,81%	0,88%	2,23%
NMS	0,83%	0,94%	2,28%
GA	0,97%	1,04%	2,22%
PSO	0,92%	1,02%	2,23%

and out-of-sample (Table 26):

Table 26: Tracking error out-of-sample

Method	Simulated Avg. TE	Market Avg. TE	DAX Avg. TE
LM	0,84%	2,45%	3,82%
NMS	0,92%	2,53%	3,69%
GA	1,14%	2,62%	4,05%
PSO	1,12%	2,61%	3,87%

In general, all four optimization methods seem to work very well, but nevertheless there sometimes occur quite significant tracking errors during the 5 years of testing as can been seen in the appendix.
We find that the Levenberg-Marquardt method generates the smallest tracking error in-sample as well as out-of-sample for the simulated bond ETF. The second best is the Nelder-Mead Simplex algorithm, the third is the Genetic algorithm and the worst method for ETF tracking we found was the Particle Swarm Optimization method
For the DAX ETF tracking we also find that the Levenberg-Marquardt method generates the smallest tracking error in-sample, but that the Nelder-Mead Simplex algorithm performs best out-of-sample, closely followed by LM.
The detailed results including the corresponding weights of the bonds can be taken from the appendix.
We further show that the tracking error can be improved by combination of LM method and GA, in which the optimal paeameters of LM were used as start parameters for GA.

4.6 Conclusion

Within this work we propose a new approach for index tracking error minimization by pushing the correlation factor between index and ETF $\rho_{(idx,ETF)} \to 1$. We then compare the quality and robustness of four different optimization methods applied to the tracking of a stock ETF index as well as two virtually constructed bond ETFs. One main problem of passively managed ETFs is to keep the generated tracking error at a minimum. The crowfoot is that the index tracking error performance is just as good as its applied calibration method. Therefore we test these methods for two different asset classes, equity and fixed income. The equity index we choose is the DAX 30 ETF index, while for the fixed income ETFs we construct indices consisting of 10 bonds with maturity 10 years but different credit spreads, using current Euro Swap rates or market data. Then we build ETFs with minimal tracking error for these 5 years, meaning that we project a possible price development of the ETF which we want to track as close as possible by a one-time asset allocation which is done in advance. Thereafter we perform an out-of-sample test using four different optimization techniques, such as Genetic algorithms, Particle Swarm Optimization, Levenberg-Marquardt and Nelder-Mead Simplex.

The results indicate a superior performance in- and out-of-sample of the LM method in fixed income ETF tracking, followed by the NMS method, and a superior performance of the LM method for the equity ETF in-sample, but NMS to be the superior optimization method for out-of-sample equity ETF tracking, followed by the LM method.

We can also see that the GA and PSO method show worse convergence than the LM and NMS method. Further, we can see that the for in-sample as well as for out-of sample the simulated tracking error of fixed-income ETFs is smaller than the tracking error calculated from real market data. The reason for that might be that for the simulated bond ETF, the CIR model assumes normal distributed short rate which seem to converge faster in the here used optimization techniques than real world returns.

For the equity ETF we find the calculated tracking errors much larger than the fixed-income ETF tracking error. One reason for this might be the higher volatility in equity markets, which makes it more difficult to comprise a smaller set of stocks which still replicates the stock index as close as possible.

Nevertheless we find for simulated as well as for real world data the LM method superior to NMS, GA, and PSO. To sum up, we find that the Levenberg-Marquardt algorithm works best for ETF tracking for simulated and for real world data as well as its application to the target function which pushes the correlation between the index and the ETF towards 1.

4.7 Appendix

4.7.1 ETF tracking error minimization with Genetic algorithms

Number of bonds using GA: 4

Simulated Bonds:	2	3	5	10		
Weights:	36,94%	26,10%	20,67%	16,29%		
Tracking Error:	i-s: 1,21%	o-o-s: 1,24%				
Historical Bonds:	2	3	5	9		
Weights:	7,03%	59,81%	26,79%	6,37%		
Tracking Error:	i-s: 1,86%	o-o-s: 3.44%				
Dax Underlying	1	6	7	9	12	14
	15	20	21	23	25	28
Weights:	6.44%	13.56%	6.97%	14.22%	10.48%	12.75%
	4.27%	12.65%	10.24%	3%	3.99%	1.44%
Tracking Error:	i-s: 2.7%	o-o-s: 5.4%				

Table 27: Optimal asset allocation using 4 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

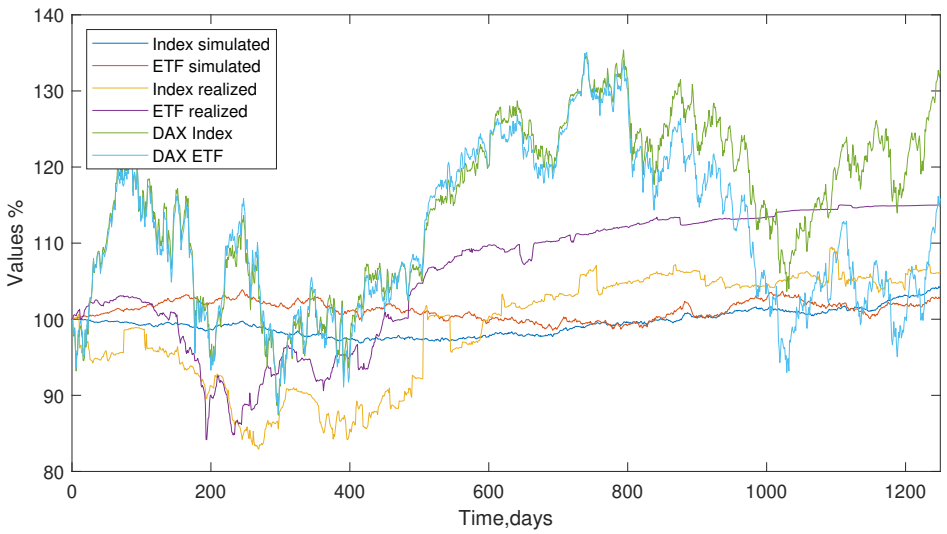

Figure 28: ETF index tracking with 4 bonds using Genetic algorithms

Number of bonds using GA: 5

Simulated Bonds:	1	2	3	5	10			
Weights:	26,61%	23,11%	15,49%	14,53%	17,27%			
Tracking Error:	i-s: 0,97%	o-o-s: 1,02%						
Historical Bonds:	2	3	7	8	9			
Weights:	26,61%	23,11%	15,49%	14,53%	17,27%			
Tracking Error:	i-s: 0,97%	o-o-s: 2,81%						
Dax Underlying	1	3	4	6	7	9	12	
	13	14	15	20	21	22	27	28
Weights:	5.46%	4%	3.39%	11.15%	5.62%	11.05%	9.24%	
	3.45%	11.08%	3.18%	9.38%	10.27%	3.89%	6.76%	2.07%
Tracking Error:	i-s: 2.24%	o-o-s: 3.26%						

Table 28: Optimal asset allocation using 5 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

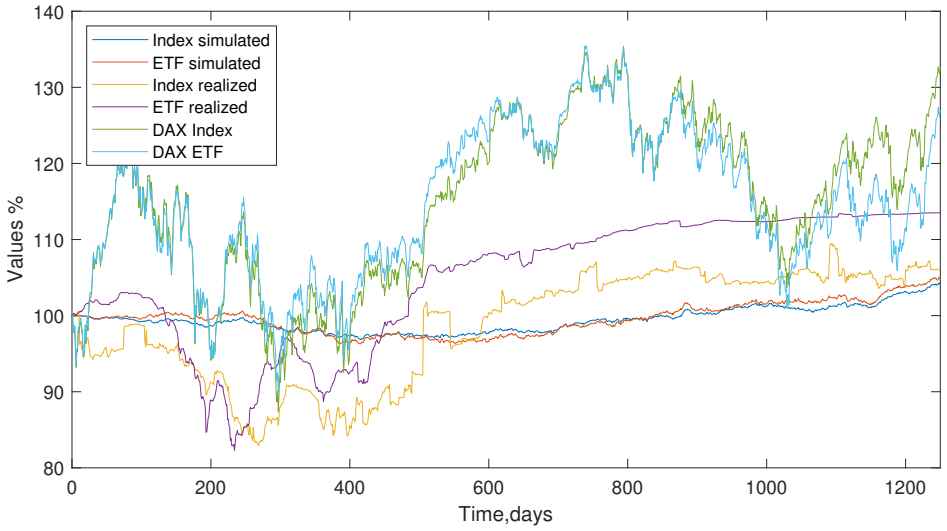

Figure 29: ETF index tracking with 5 bonds using Genetic algorithms

Number of bonds using GA: 6

Simulated Bonds:	1	2	3	5	6	7		
Weights:	24,94%	18,84%	12,31%	11,33%	14,07%	18,50%		
Tracking Error:	i-s: 0,74%	o-o-s: 0,76%						
Historical Bonds:	1	2	3	4	5	10		
Weights:	22,49%	26,72%	24,19%	7,76%	10,63%	8,21%		
Tracking Error:	i-s: 0,67%	o-o-s: 2,25%						
Dax Underlying	1	3	4	6	8	9	12	14
	15	16	17	20	21	23	25	26
	27	30						
Weights:	4.43%	3.44%	3.34%	9.38%	2.41%	8.23%	9.56%	10.14%
	3.21%	2.97%	4.41%	9.86%	10.34%	3.35%	2.9%	3.83%
	7.29%	0.91%						
Tracking Error:	i-s: 2.03%	o-o-s: 3.58%						

Table 29: Optimal asset allocation using 6 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

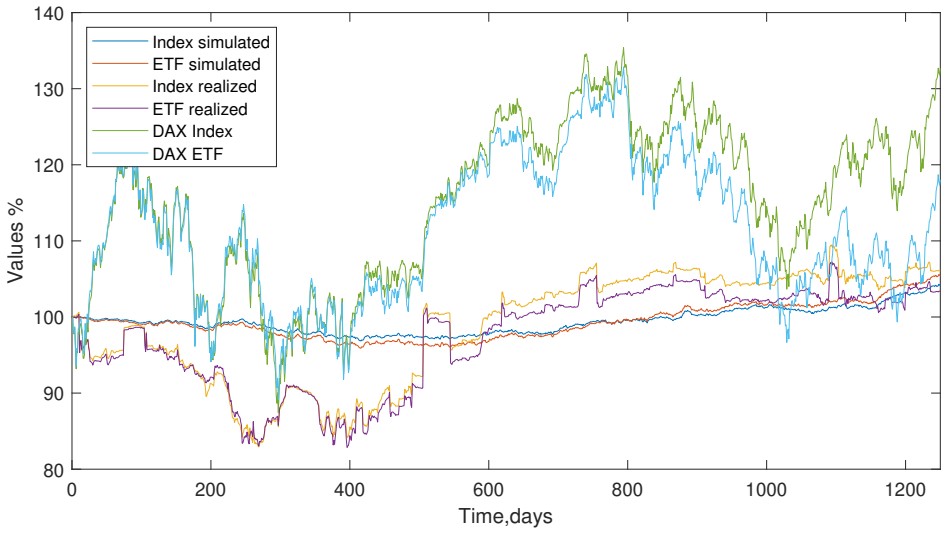

Figure 30: ETF index tracking with 6 bonds using Genetic algorithms

Number of bonds using GA: 7

Simulated Bonds:	1	2	3	4	5	6	7
Weights:	22,64%	17,30%	11,32%	8,03%	10,57%	13,05%	17,09%
Tracking Error:	i-s: 0,33%	o-o-s: 0,35%					
Historical Bonds:	1	2	3	4	5	8	10
Weights:	20,58%	21,79%	17,50%	7,10%	9,81%	15,62%	7,59%
Tracking Error:	i-s: 0,64%	o-o-s: 1,98%					
Dax Underlying	1	3	4	5	6	7	9
	11	12	13	14	15	16	17
	20	21	23	24	25	26	27
Weights:	4.26%	3.17%	1.62%	0.86%	9.27%	4.45%	8.19%
	2.14%	9.16%	2.94%	9.06%	3.72%	4.05%	1.98%
	8.23%	9.19%	3.53%	0.33%	3.46%	4.33%	6.05%
Tracking Error:	i-s: 1.89%	o-o-s: 3.95%					

Table 30: Optimal asset allocation using 7 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

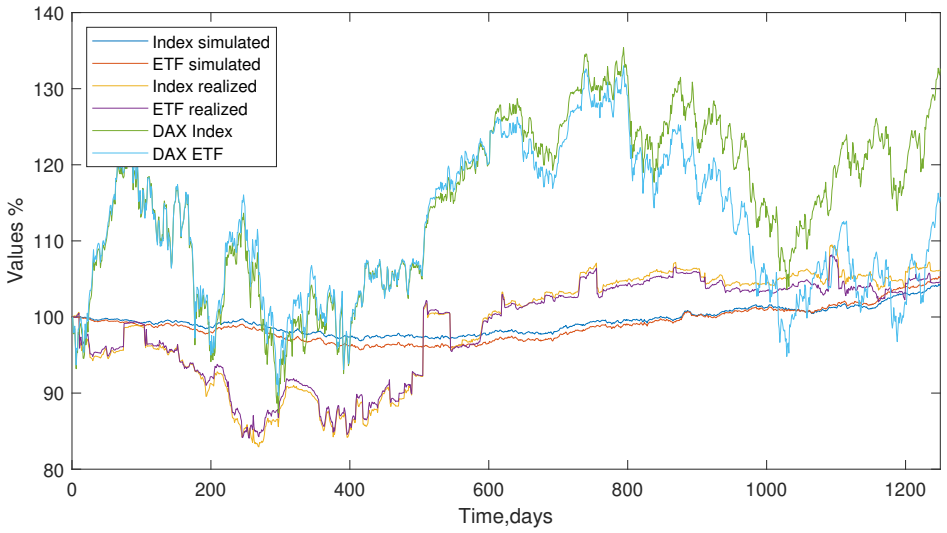

Figure 31: ETF index tracking with 7 bonds using Genetic algorithms

4.7.2 ETF tracking error minimization with Particle Swarm Optimization

Number of bonds using PSO: 4

Simulated Bonds:	2	3	5	10		
Weights:	36,93%	26,10%	20,67%	16,28%		
Tracking Error:	i-s: 1,21%	o-o-s: 1,24%				
Historical Bonds:	2	3	5	9		
Weights:	6,6%	60,09%	26,92%	6,40%		
Tracking Error:	i-s: 1,32%	o-o-s: 3,25%				
Dax Underlying	3	6	12	13	14	15
	19	20	21	25	27	29
Weights:	3.96%	13.42%	11.9%	5.04%	13.14%	3.3%
	9.41%	11.44%	12.17%	4.75%	9.76%	1.7%
Tracking Error:	i-s: 2.78%	o-o-s: 3.91%				

Table 31: Optimal asset allocation using 4 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

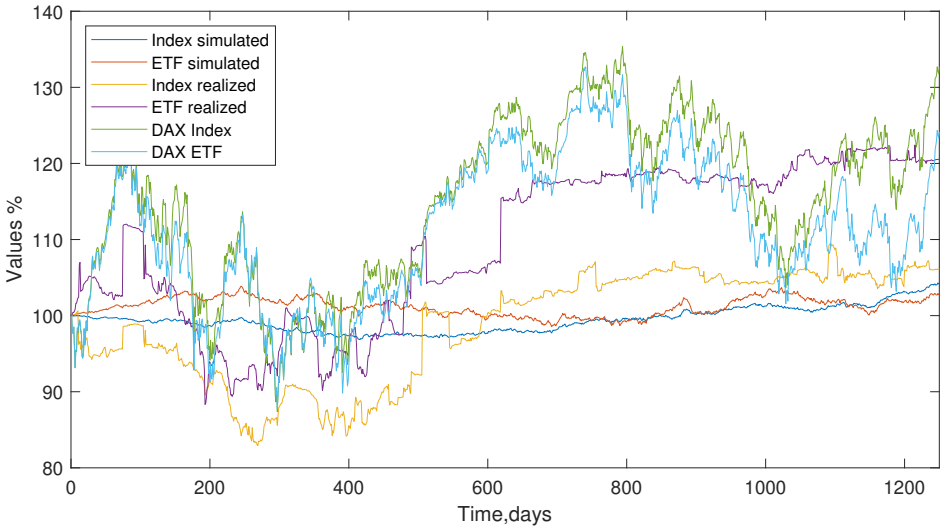

Figure 32: ETF index tracking with 4 bonds using Particle Swarm Optimization

Number of bonds using PSO: 5

Simulated Bonds:	1	2	3	5	6			
Weights:	29,62%	23,10%	15,48%	14,54%	17,26%			
Tracking Error:	i-s: 0,97%	o-o-s: 1,02%						
Historical Bonds:	4	7	8	9	10			
Weights:	17,41%	60,23%	0,01%	5,63%	16,72%			
Tracking Error:	i-s: 1,41%	o-o-s: 2,86%						
Dax Underlying	4	6	7	9	12	14	15	
	17	19	20	21	24	27	28	30
Weights:	3.19%	11.59%	4.28%	9.43%	9.29%	12.14%	3.34%	
	2.29%	7.63%	11.84%	11.03%	1.99%	7.68%	2.26%	2.01%
Tracking Error:	i-s: 2.47%	o-o-s: 3.69%						

Table 32: Optimal asset allocation using 5 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

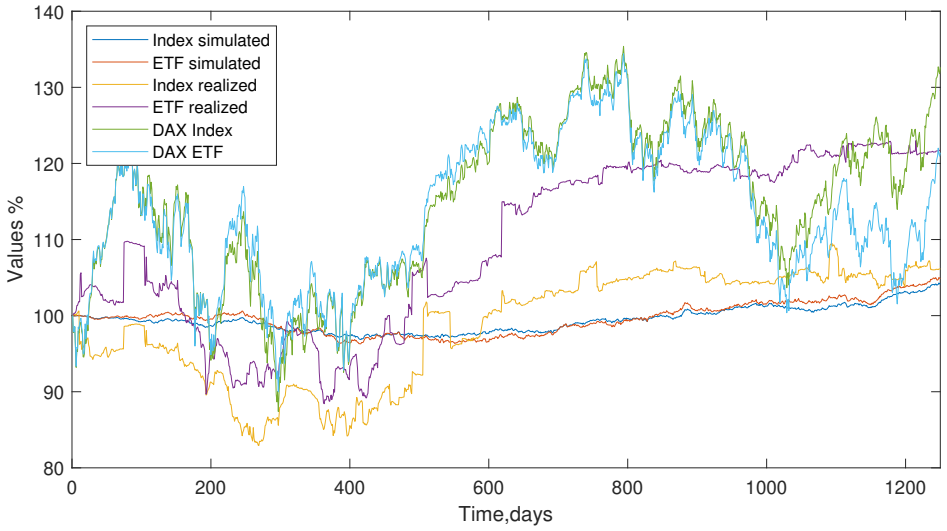

Figure 33: ETF index tracking with 5 bonds using Particle Swarm Optimization

Number of bonds using PSO: 6

Simulated Bonds:	1	2	3	5	6	7		
Weights:	24,95%	18,84%	12,31%	11,34%	14,07%	18,50%		
Tracking Error:	i-s: 0,72%	o-o-s: 0,73%						
Historical Bonds:	1	2	3	4	5	10		
Weights:	23,08%	24,81%	24,82%	7,96%	10,89%	8,43%		
Tracking Error:	i-s: 0,68%	o-o-s: 2,31%						
Dax Underlying	1	3	4	6	8	9	12	14
	15	16	17	20	21	23	25	26
	27	30						
Weights:	4.35%	3.35%	2.81%	9.71%	2.26%	8.44%	9.58%	10.27%
	2.95%	3.12%	4.35%	10.01%	10.28%	3.5%	3.16%	3.92%
	7.02%	0.94%						
Tracking Error:	i-s: 2.03%	o-o-s: 3.71%						

Table 33: Optimal asset allocation using 6 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

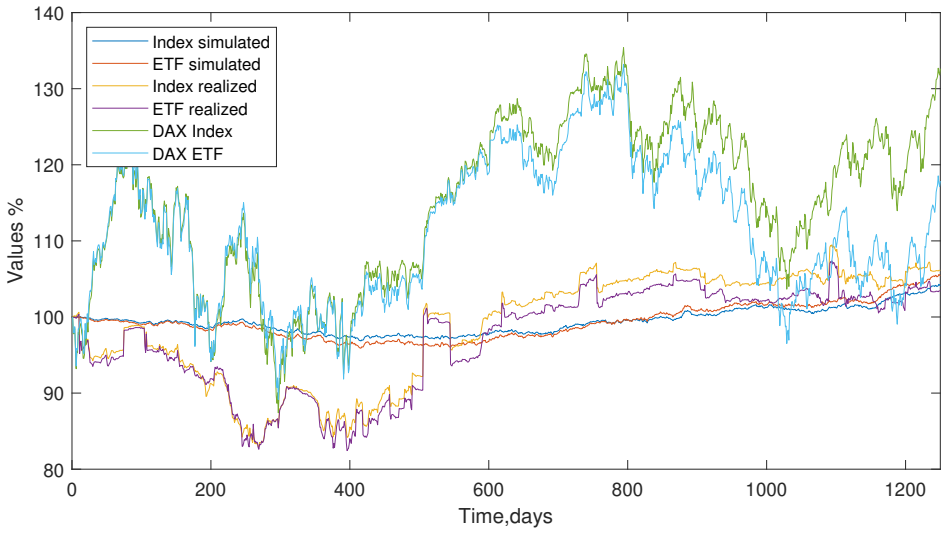

Figure 34: ETF index tracking with 6 bonds using Particle Swarm Optimization

Number of bonds using PSO: 7

Simulated Bonds:	1	2	3	4	5	6	7
Weights:	22,67%	17,31%	11,28%	8,03%	10,59%	13,01%	17,10%
Tracking Error:	i-s: 0,33%	o-o-s: 0,35%					
Historical Bonds:	1	2	3	4	5	8	10
Weights:	20,76%	21,16%	17,40%	7,22%	9,89%	15,96%	7,61%
Tracking Error:	i-s: 0,65%	o-o-s: 2,01%					
Dax Underlying	3	4	6	7	9	12	13
	14	16	18	19	20	21	22
	23	24	25	26	27	29	30
Weights:	2.82%	2.93%	9.77%	4.87%	6.79%	9.56%	3.52%
	10.75%	3.42%	2.14%	3.23%	7.61%	9.61%	3.29%
	3.11%	0.57%	2.41%	3.73%	7.55%	1.24%	1.09%
Tracking Error:	i-s: 1.86%	o-o-s: 3.47%					

Table 34: Optimal asset allocation using 7 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

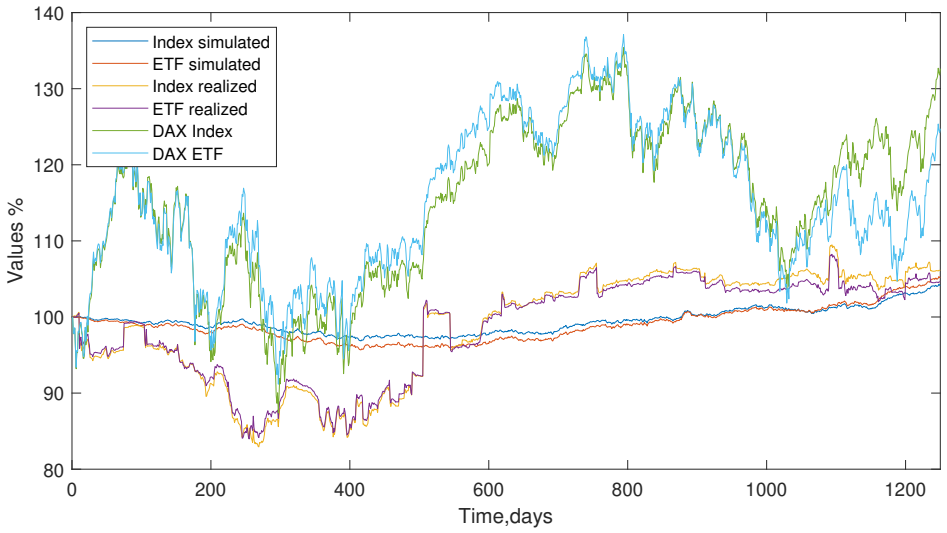

Figure 35: ETF index tracking with 7 bonds using Particle Swarm Optimization

4.7.3 Fixed-income ETF tracking error minimzation with Levenberg-Marquardt

Number of bonds using LM: 4

Simulated Bonds:	2	3	5	10		
Weights:	36,93%	26,09%	20,69%	16,29%		
Tracking Error:	i-s: 1,21%	o-o-s: 1,24%				
Historical Bonds:	4	7	8	9		
Weights:	20,20%	67,06%	5,76%	6,99%		
Tracking Error:	i-s: 1,26%	o-o-s: 3.10%				
Dax Underlying	6	7	9	12	14	16
	17	20	21	23	25	27
Weights:	11.93%	3.52%	14.18%	9.8%	12.78%	4.47%
	3.22%	12.26%	11.44%	4.03%	4.54%	7.83%
Tracking Error:	i-s: 2.67%	o-o-s: 4.8%				

Table 35: Optimal asset allocation using 4 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

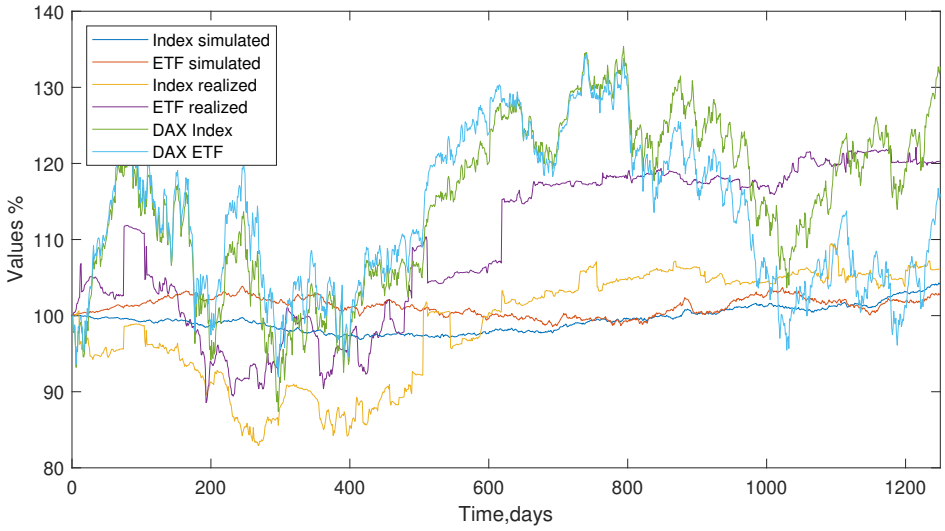

Figure 36: ETF index tracking with 4 bonds using Levenberg-Marquardt

Number of bonds using LM: 5

Simulated Bonds:	1	2	3	5	6			
Weights:	29,56%	23,10%	15,50%	14,56%	17,28%			
Tracking Error:	i-s: 0,97%	o-o-s: 1,02%						
Historical Bonds:	2	3	8	9	10			
Weights:	21,08%	32,56%	28,91%	4,69%	12,77%			
Tracking Error:	i-s: 0,80%	o-o-s: 2,52%						
Dax Underlying	3	4	6	8	9	12	13	
	14	15	17	20	21	22	25	26
Weights:	3.76%	3.64%	9.86%	3.35%	9.98%	10.83%	4.92%	
	11.92%	3.8%	4.87%	11.38%	10.56%	3.82%	3.63%	3.71%
Tracking Error:	i-s: 2.38%	o-o-s: 3.81%						

Table 36: Optimal asset allocation using 5 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

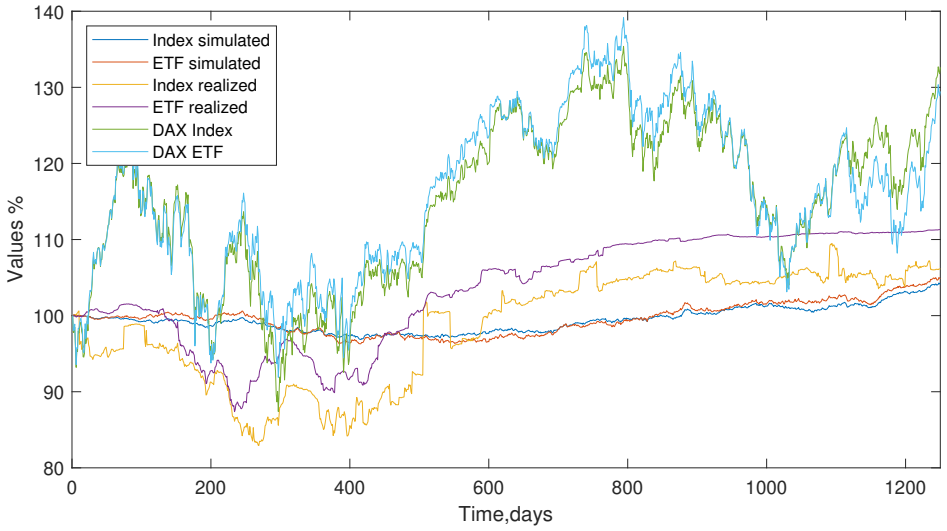

Figure 37: ETF index tracking with 5 bonds using Levenberg-Marquardt

Number of bonds using LM: 6

Simulated Bonds:	1	2	3	5	6	7		
Weights:	24,94%	18,84%	12,31%	11,33%	14,07%	18,50%		
Tracking Error:	i-s: 0,72%	o-o-s: 0,73%						
Historical Bonds:	2	3	5	6	9	10		
Weights:	18,38%	38,81%	18,09%	6,75%	3,88%	14,08%		
Tracking Error:	i-s: 0,80%	o-o-s: 2,20%						
Dax Underlying	1	2	3	6	9	11	12	13
	14	15	17	19	20	21	22	27
	28	29						
Weights:	4.29%	2.46%	3.24%	10.96%	8.13%	3.69%	9.34%	4.43%
	10.17%	2.89%	4.3%	3.92%	8.81%	9.69%	3.7%	6.99%
	1.52%	1.48%						
Tracking Error:	i-s: 2.1%	o-o-s: 3.09%						

Table 37: Optimal asset allocation using 6 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

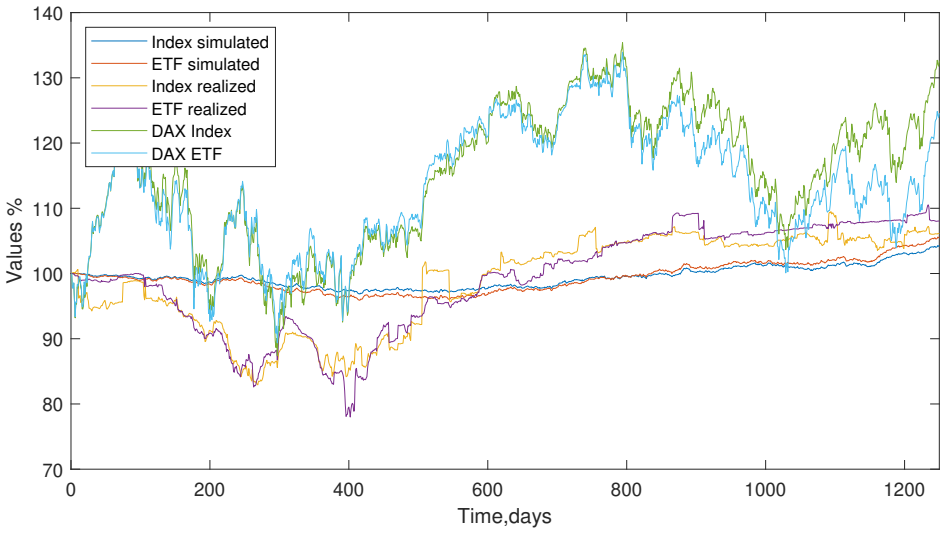

Figure 38: ETF index tracking with 6 bonds using Levenberg-Marquardt

Number of bonds using LM: 7

Simulated Bonds:	1	2	3	4	5	6	7
Weights:	22,67%	17,31%	11,28%	8,03%	10,60%	13,01%	17,10%
Tracking Error:	i-s: 0,33%	o-o-s: 0,35%					
Historical Bonds:	1	2	3	4	5	6	8
Weights:	21,89%	14,44%	12,10%	7,67%	9,90%	13,30%	20,71%
Tracking Error:	i-s: 0,64%	o-o-s: 1,96%					
Dax Underlying	1	3	4	6	7	9	12
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
Weights:	4.15%	2.88%	2.87%	9.72%	3.86%	6.61%	8.96%
	10%	2.37%	3.32%	2.41%	1.03%	3.03%	9.13%
	9.68%	3.23%	3.38%	0.45%	2.68%	3.56%	6.69%
Tracking Error:	i-s: 1.77%	o-o-s: 3.57%					

Table 38: Optimal asset allocation using 7 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

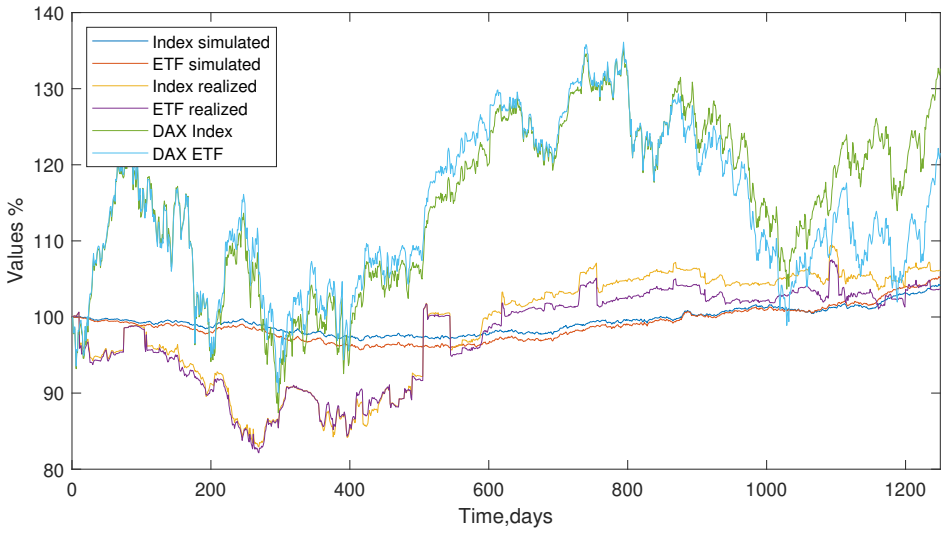

Figure 39: ETF index tracking with 7 bonds using Levenberg-Marquardt

4.7.4 ETF tracking error minimzation with Nelder-Mead Simplex (NMS)

Number of bonds using NMS: 4

Simulated Bonds:	2	3	5	10		
Weights:	36,94%	26,10%	20,67%	16,29%		
Tracking Error:	i-s: 1,21%	o-o-s: 1,24%				
Historical Bonds:	2	3	5	9		
Weights:	6,68%	60,03%	26,89%	6,39%		
Tracking Error:	i-s: 1,35%	o-o-s: 3.35%				
Dax Underlying	3	5	6	7	8	9
	12	14	19	20	21	23
Weights:	3.83%	3.31%	13.89%	6.84%	6.02%	8.97%
	10.24%	13.94%	5%	12.62%	11.26%	4.08%
Tracking Error:	i-s: 2.72%	o-o-s: 4.43%				

Table 39: Optimal asset allocation using 4 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

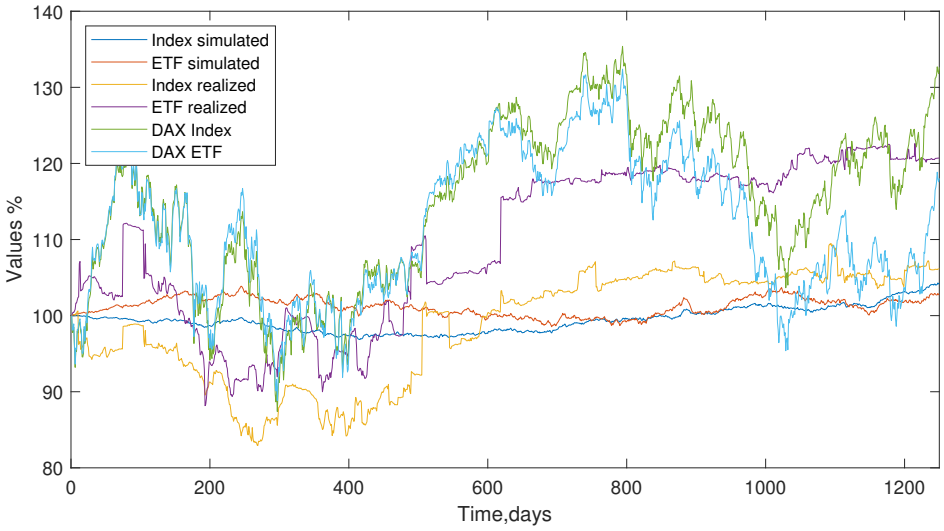

Figure 40: ETF index tracking with 4 bonds using Nelder-Mead Simplex

Number of bonds using NMS: 5

Simulated Bonds:	1	2	3	5	6			
Weights:	29,61%	23,11%	15,48%	14,53%	17,27%			
Tracking Error:	i-s: 0,97%	o-o-s: 1,02%						
Historical Bonds:	1	2	3	4	5			
Weights:	22,51%	35,33%	23,86%	7,75%	10,54%			
Tracking Error:	i-s: 0,73%	o-o-s: 2,65%						
Dax Underlying	1	3	4	6	7	9	12	
	14	17	20	21	22	24	25	27
Weights:	5.29%	4.34%	3.08%	10.31%	3.16%	10.62%	9.23%	
	12.25%	3.22%	12.3%	11.11%	3.25%	0.45%	3.66%	7.73%
Tracking Error:	i-s: 2.33%	o-o-s: 3.89%						

Table 40: Optimal asset allocation using 5 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

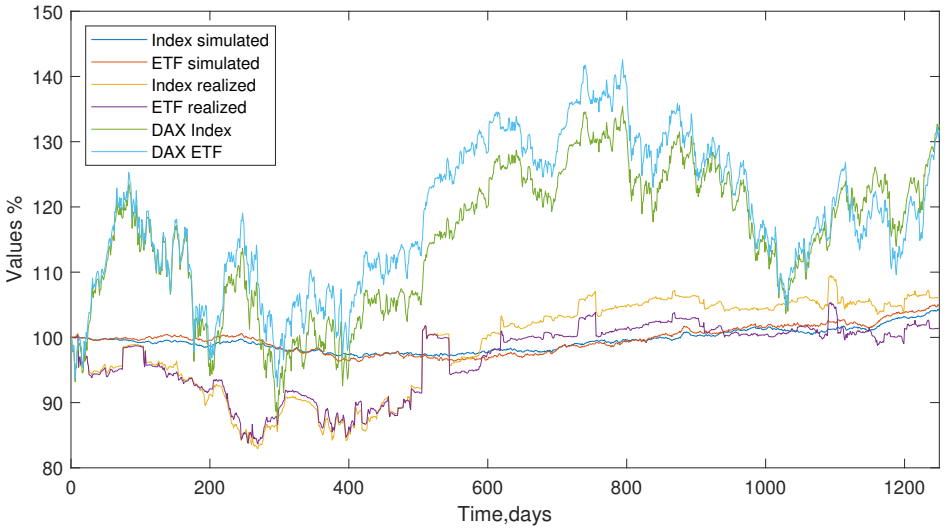

Figure 41: ETF index tracking with 5 bonds using Nelder-Mead Simplex

Number of bonds using NMS: 6

Simulated Bonds:	1	2	3	5	6	7		
Weights:	24,95%	18,84%	12,31%	11,34%	14,07%	18,50%		
Tracking Error:	i-s: 0,72%	o-o-s: 0,73%						
Historical Bonds:	1	2	3	4	5	10		
Weights:	21,17%	31,04%	22,75%	7,30%	10,01%	10%		
Tracking Error:	i-s: 0,63%	o-o-s: 2,11%						
Dax Underlying	1	2	3	6	7	9	10	12
	14	15	19	20	21	22	23	26
	27	28						
Weights:	4.7%	2.3%	2.87%	12.1%	5.65%	6.83%	-0.04%	9.17%
	10.45%	3.48%	2.9%	9.76%	10.1%	3.85%	3.14%	4.38%
	6.9%	1.47%						
Tracking Error:	i-s: 2.03%	o-o-s: 3.78%						

Table 41: Optimal asset allocation using 6 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

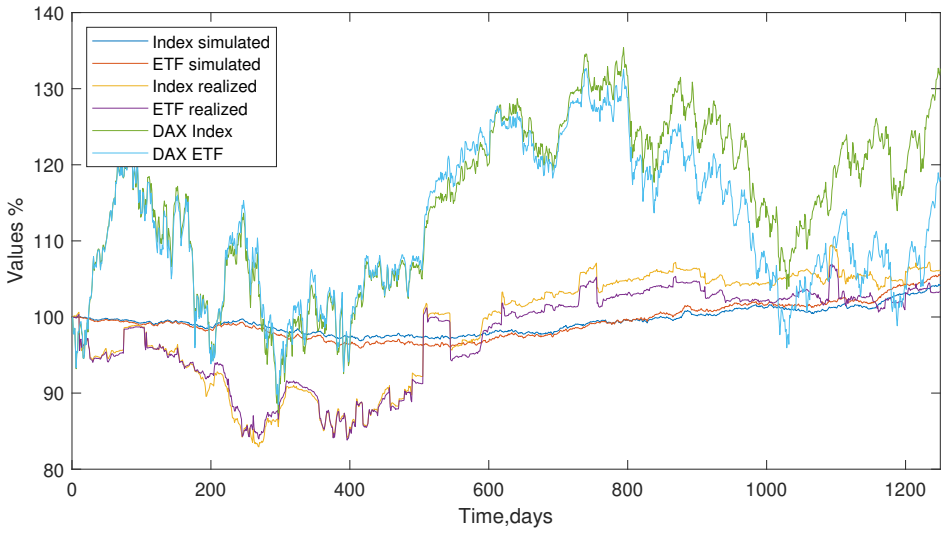

Figure 42: ETF index tracking with 6 bonds using Nelder-Mead Simplex

Number of bonds using NMS: 7

Simulated Bonds:	1	2	3	4	5	6	7
Weights:	22,67%	17,31%	11,28%	8,03%	10,60%	13,01%	17,10%
Tracking Error:	i-s: 0,33%	o-o-s: 0,35%					
Historical Bonds:	1	2	3	4	5	8	10
Weights:	20,59%	21,80%	17,26%	7,16%	9,81%	15,82%	7,55%
Tracking Error:	i-s: 0,65%	o-o-s: 2%					
Dax Underlying	1	2	3	4	5	6	7
	9	11	12	14	15	17	19
	20	21	22	23	26	27	29
Weights:	3.57%	1.83%	2.97%	2.51%	1.41%	10.41%	3.54%
	6.84%	2.73%	8.95%	10.69%	3.35%	2.3%	2.49%
	8.64%	9.76%	3.54%	2.93%	3.92%	6.46%	1.16%
Tracking Error:	i-s: 1.83%	o-o-s: 3.39%					

Table 42: Optimal asset allocation using 7 bonds for replicating the index, corresponding weights and resulting tracking error in-sample and out-of-sample.

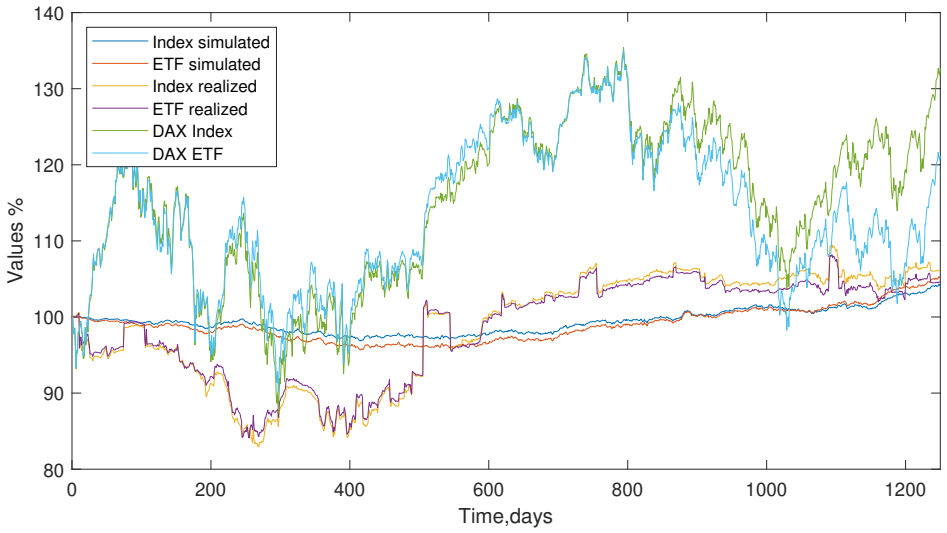

Figure 43: ETF index tracking with 7 bonds using Nelder-Mead Simplex

4.7.5 Euro Swaps Curve

Date	Yield
1D	-0.052
1W	-0.01
1M	0.022
2M	0.048
3M	0.082
6M	0.179
7M	0.161
8M	0.151
9M	0.151
10M	0.154
11M	0.166
1Y	0.179
18M	0.19
2Y	0.207
3Y	0.259
4Y	0.326
5Y	0.403
6Y	0.5
7Y	0.61
8Y	0.728
9Y	0.844
10Y	0.952
11Y	1.051
12Y	1.138
15Y	1.34
20Y	1.531
25Y	1.615
30Y	1.66
35Y	1.687
40Y	1.706
45Y	1.709
50Y	1.698

Table 43: Euro Swaps Curve 08/12/14.

References

[1] Brian C. Blanchett and David M. Blanchett, Tracking Tracking Error, *Journal of Indexes*, June 18, 2007, http://www.etf.com/publications/journalofindexes/joi-articles/2870.html.

[2] Kennedy J. and Eberhart R., Particle Swarm Optimization, *Proceedings of the Fourth IEEE International Conference on Neural Networks*, Perth, Australia. IEEE Service Center(1995) 1942-1948.

[3] Roland Jeurissen and Jan van den Berg, Index Tracking using a Hybrid Genetic Algorithm, *Computational Intelligence Methods and Applications*, 2005 ICSC Congress, 10.1109/CIMA.2005.1662364.

[4] H.H.N. Amin, Calibration of Different Interest Rate Models for a Good Fit of Yield Curves, Thesis, *Delft University of Technology Faculty of Electrical Engineering*, Mathematics and Computer Science Delft Institute of Applied Mathematics, September 2012.

[5] Monika Piazzesi, Affine Term Structure Models, *Handbook of Financial Econometrics*, 2010, Vol. 1, Chapter 12, 697-700.

[6] John Y. Campbell, A Defense of Traditional Hypotheses about the Term Structure of Interest Rates, *The Journal of Finance*, Vol. 41, No. 1 (Mar., 1986), pp. 183-193.

[7] Y.-S. Kim, Refined Simplex Method for Data Fitting, *Astronomical Data Analysis Software and Systems VI*, ASP Conference Series, Vol. 125, 1997.

[8] ETF Specific Data Point Methodologies, Morningstar,*Morningstar Methodology Paper*, December 31, 2010.

[9] S.C.Wong, C.K.Wong and C.O.Tong, A parallelized genetic algorithm for the calibration of Lowry model, *Parallel Computing*, 27: 1523-1536, 2001.

[10] Marquardt, D. W., An Algorithm for Least Squares Estimation of Nonlinear Parameters, *Journal Society Industrial Applied Mathematics*, Vol. 11, No. 2, pp. 431-441, June 1963.

[11] Evelyn Araneda, A Variation of the Levenberg Marquardt Method. An attempt to improve efficiency. *Submitted to the Department of Earth, Atmospheric and Planetary Sciences in Partial Fulfillment for the Master of Science in Geosystems*, MIT, May, 2004.

[12] Jörg Kienitz and Daniel Wetterau, Financial modelling, Theory, Implementation and Practice with Matlab Source, *John Wiley & Sons Ltd*, p. 437, 2012.

[13] Thirumalai, Ramabhadran S., Active vs. passive ETF, Indiana University, Bloomington, 2003

5 Fuel hedging in an inflated environment

Kujtim Avdiu [1] and Stephan Unger[2]

[12] Department of Finance, Faculty of Business, Economics and Statistics,University of Vienna
Brünner Strasse 72, 1210 Vienna, Austria
e-mail: kujtim.avdiu@univie.ac.at
e-mail: stephan.unger@univie.ac.at

February 2013

Abstract

This paper proposes an option pricing application for the hedging of fuel prices. We consider a world where all commodities have the same expected return, defined by the inflation rate. Therefore the risk-free change in fuel price is assumed to follow the inflation rate. Since fuel can be considered as a derivative of oil, it is interesting to notice that there does not exist a hedging possibility for the end product after refinement, transportation and taxes. Since the amount of intermediary trades between the raw and the end product influences the end price, many consumers are exposed to the end price. Considering the observations of the end price as the filtration being generated, they allow us to apply a hedging formula to the fuel prices in form of an American call option. Industries consider different ways of hedging their consumption exposure with crude oil futures, correlations between crude oil and kerosine, and so on. But for companies and individuals who are highly exposed to road traffic based vehicles such as cars and trucks, it would be of advantage to have a direct hedging instrument such as a fuel option.

Keywords: Fuel price, fuel consumption, inflation, American call option, binomial option pricing.

Author contribution
The basic idea to develop a model for hedging of fuel prices came from S.U. K.A. proposed a binomial model after doing an extensive literature review. S.U. provided the data and K.A. performed the calculations and programming. S.U. & K.A. interpreted and summarized the results together and wrote the conclusion.

Published: Journal of Financial and Economic Practice; Peoria, Volume 13 Issue 2, (Fall 2013): 0_3

5.1 Introduction

Fuel consumption is a driving factor of the economy. Price fluctuations in commodity prices influence global economic growth and are characterized by cyclical phases of up and down movements. Since the general demand in fuel is strongly correlated to economic growth, we face a rapid increase in fuel prices due to several different factors.

First, far Eastern countries record a high economic growth which leads to a high demand in fuel. This again causes a steady increase in commodity prices in general, especially in fuel.

A second point is related to the supply-demand problem which is currently influencing fuel price determination. It originates in the uncertainty about future fuel supply due to a contingent shortfall of fuel export in Middle East. Markets rumour and steady uncertainty about such issues heat prices and make them very sensitive to any kind of news related to them. The effect we observe is higher volatility which in turn makes prices more expensive due to a higher risk premium.

A third point that stresses the price increases in fuel is the current imbalance in national budgets. In order to reduce the high budget deficits European governments try to get rid off its debts by increasing inflation in long-term. Indicators for such behaviour are programs such as the CBPP (covered bond purchase program) 1 and 2 indicated by the European Central Bank. In order to stabilize financial markets by buying goverment bonds across Europe, the negative long-term effect is that money supply M0 inevitably increases too. In case of affecting M3, this will lead to higher inflation. As many European economies are struggeling, the increased money supply has no negative effect at the moment. But one has to expect that as soon as economies recover and spending power increases, fuel prices will tend to increase due to higher demand in crude oil which in turn heats inflation.

The combination of these factors seems to have a toxic effect on the natural business cycle. There are limited ways of hedging against increasing fuel prices. Industrial sectors which heavily depend on oil prices such as airlines hedge by use of standard market instruments such as option structures or futures in crude oil. But due to the natural imperfections in hedging, increasing oil prices will also lead to higher costs for these industries.

The goal of this paper is to introduce a hedging proposal for increasing fuel prices with respect to the road traffic. Since this sector is widely forgotten by financial product development, this paper proposes a hedging approach on fuel by pricing an American option on fuel which may be exercised directly at the gas station. The reason for the lack of interest in developing hedging opportunities for this area may stem from the fact that road traffic is no industry sector in classic terms such as

the airline sector, whereas it affects any business which is conducted by road traffic from cargo companies to private car drivers.

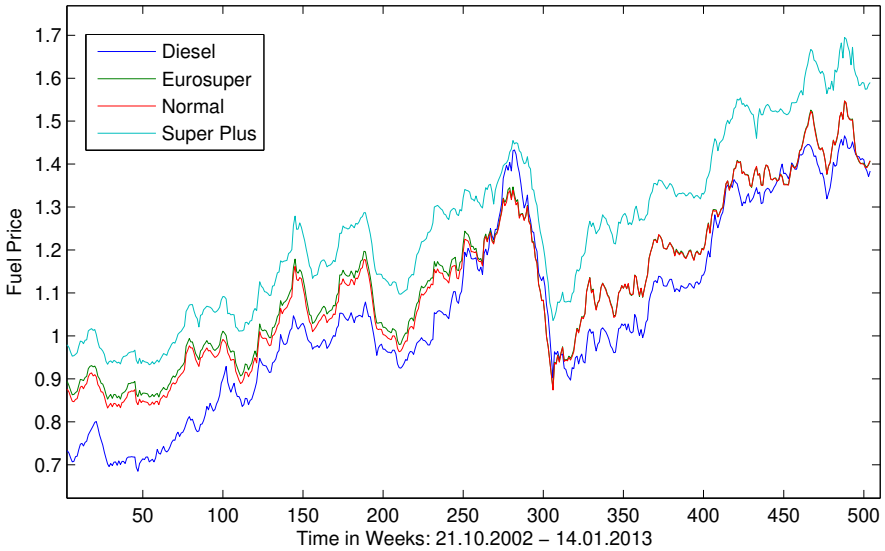

Figure 1: Historical fuel price development (diesel (blue), Eurosuper unleaded (green), normal 90 octane (red) and super plus (light blue))[6]

5.2 The model

The model proposed here is based on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, \mathcal{F}_{0\leq t<\infty})$, where price processes are assumed to be $(\mathcal{F}_t)$-adapted. The underlying price process we deal with can be described by the change in the fuel price dS_t. The fuel price dynamics (under a real-world probability measure $\mathbb{P}$) are assumed to take the form

$$\frac{dS_t}{S_t} = \mu dt + \sigma(t, S) dW_t, \tag{129}$$

where the processes $\mu(t)_{0\leq t<\infty}$ and $\sigma(t)_{0\leq t<\infty}$ are $\mathcal{F}_t$-adapted, measurable and uniformly bounded. Moreover the fuel price satisfies the no-arbitrage condition. Assuming that all future prices are observable leads to the conclusion that also the spot price is observable. Therefore absence of arbitrage implies

$$\mathbb{E}_t^{\widetilde{\mathbb{P}}}[dS_t] = (r_t^N - F_t) S(t) dt, \tag{130}$$

where r_t^N is the nominal risk-free rate and F_t is the instantaneous convenience yield. Therefore the fuel price is assumed to take the form

$$S_t = S_0 e^{Y_t}, \tag{131}$$

where (Y_t) is taken to be a $\mathbb{P}$-Brownian motion with drift. Thus for each t in $[0, \inf)$ we have

$$Y_t \sim N(\mu t, \sigma^2 t). \tag{132}$$

Assuming that S_t represents the value of the fuel price at time t, and Z_t represents the value of an American call option at time t, we are interested in the relation of the prices at the various exercise times.

5.3 Pricing a Fuel option

5.3.1 Continuous time case

The time interval in which the model is studied is a bounded interval [0,T]. We consider next an American call option with maturity T, for which its payoff is characterized by the non-negative adapted continuous process $Z = (Z_t)_{0 \leq t \leq T}$. What we are interested in is a hedge against raising fuel prices. Therefore the payoff of the American call option is

$$Z_t = (S_t - K)_+. \tag{133}$$

For integrability we need

$$\mathbb{E}^* sup_{0 \leq t \leq T} Z_t < \infty. \tag{134}$$

Definition 3.1: A replicating payoff for the random fuel price process value S_t is any admissible payoff of the American option with value $Z = (Z_t)_{0 \leq t \leq T}$ such that, with probability one,

$$\forall t \in [0,T],\ Z_t \geq S_t. \tag{135}$$

One characteristic of hedging fuel is that of early exercise. As it would not make sense to price an option of European-style e.g. for car drivers on a long-term basis, we need to take into account all possible exercise dates the fuel option may be exercised. Therefore we need to maximize $\mathbb{E}[Z_\nu]$ over all finite stopping times ν, with $Z = (Z_t)_{t \in \mathbb{N}}$ being an $\mathcal{F}_t$-adapted sequence of random variables with $\mathbb{E} sup_{t \in \mathbb{N}} |Z_t| < \infty$. In the setting of American options, Z_t represents the hedging profit attachted to exercizing the option at time t.

Let τ be the set of all stopping times with respect to the filtration $(\mathcal{F}_t)_{t \in \mathbb{N}}$. Then

$$\tau_{t,T} = \{\nu \in \tau \quad | \mathbb{P}(\nu) \in [t, T]) = 1\},\ 0 \leq t \leq T, \tag{136}$$

$$\tau_{t,\infty} = \{\nu \in \tau \quad | \mathbb{P}(\nu) \in [t, +\infty]) = 1\},\ t \in T. \tag{137}$$

Definition 3.2: The Snell envelope $\tilde{U}$, of $(Z_t)_{t\in\mathbb{N}}$ is the sequence $(U_t)_{t\in\mathbb{N}}$ defined by

$$U_t{=}ess \quad sup_{\nu\in\tau_{t,\infty}}\mathbb{E}[Z_\nu|\mathcal{F}_t]. \tag{138}$$

Suppose that $(\tilde{V}_t)_{0\leq t\leq T}$ is the discounted value of a car driver's hedging strategy and $V{=}\,(V_t)_{0\leq t\leq T}$ is the value process of this strategy such that, with probability one,

$$\forall t{\in}[t,T],\, V_t{\geq}Z_t. \tag{139}$$

Due to the fact that under $\widetilde{\mathbb{P}}$ the discounted value of an admissible strategy is a supermartingale, we can set up following proposition:

Proposition 3.1: Consider the price process of an American option defined by a nonnegative, continuous, adapted process $Z{=}\,(Z_t)_{0\leq t\leq T}$, which satisfies (134). Denote by $\tilde{U}$ the Snell envelope, under $\widetilde{\mathbb{P}}$, of the process $\tilde{Z}$, with $\tilde{Z}_t{=}\beta_t Z_t$ and let U be the process defined by $U_t{=}S_t^0\tilde{U}_t$, where β represents the discount factor. We have

$$U_t{=}ess \quad sup_{\tau\in[t,T]}\mathbb{E}^{\widetilde{\mathbb{P}}}(e^{-\int_t^\tau r_s ds}Z_\tau|\mathcal{F}_t), \tag{140}$$

and, if V is the value process of any replicating strategy for the American option, we have, almost surely, $V_t{\geq}U_t$, for every $t{\in}[0,T]$. [3]

Proof: see [3].

5.3.2 Discrete time case

For the purpose of application, the pricing of a fuel option in a discrete time frame might be more convenient and even more sufficient. Therefore we construct a binomial tree with T time steps corresponding to times $k{=}\,0,1,...,T$, which models the fuel price S_k. The time T corresponds to the time horizon a car driver would like to hedge his fuel consumption. In our framework we assume a possible early exercise of the American call option every day.

We make use of the notation of Mark Davis as in [2]. At each node, resp. every day, the fuel price moves either up to $S_{k+1}{=}uS_k$ or down to $S_{k+1}{=}dS_k$ with $u{>}\,1$ and $d{=}\frac{1}{u}$. For each up and down movement we assume the same probability of occurance. We let $\mathcal{F}_k$ be the σ-field generated by $\{S_0, S_1, ..., S_k\}$. At time k the possible price values are specified by a vector $s_k{=}s_k[0],...,s_k[k]$ with

$$s_k[j]\,{=}u^k d^{2j}{=}d^{2j-k}. \tag{141}$$

In a risk-neutral environment any saved fuel is worth e^r at time T. The up and down probabilities p and d are risk-neutral if the discounted price $\frac{S_k}{r^k}$ is a martingale. Therefore, at time 0 this requires that $S_0=\frac{1}{e^r}(pu+(1-p)d) =\mathbb{E}[\frac{S_1}{r}]$, a condition that is required at every node. Hence

$$p=\frac{e^r-d}{u-d}, \tag{142}$$

$$d=\frac{e^r-u}{d-u}. \tag{143}$$

In case of the American call option with strike K, reflecting the fuel price, the buyer has the right to buy fuel with exercise value $[S_k-K]^+$. The pricing algorithm works backwards and takes into account every exercise possibility by discounting every payoff at every node. For this reason we consider a general exercise value $\hat{Y}_{kj}$. The function $\tilde{Z}$ is defined by

$$\tilde{Z}_{Nj}=\hat{Y}_{Nj}, j= 0, ...,N, \tag{144}$$

$$\tilde{Z}_{(k-1)j}=max\{\hat{Y}_{(k-1)j}, \frac{1}{e^r}[p\tilde{Z}_{kj}+(1-p)\tilde{Z}_{k(j+1)}]\},j= 0, ...,k-1,k=T, ..., 1. \tag{145}$$

Next we denote S as the stopping region $S=\{(k, j) :\hat{Z}_{kj}=\hat{Y}_{kj}\}$. Now let τ^* be a random stopping time

$$\tau^*=min\{k:\hat{Z}(k, J_k) =\hat{Y}(k, J_k)\}=min\{k: (k, J_k)\in S\}. \tag{146}$$

At time N we have $\hat{Z}=\hat{Y}$.

Proposition 3.2:

1. $\hat{Z}_{00}=max_{\tau\in\sum}\mathbb{E}[r^{-\tau}\hat{Y}(\tau, J_\tau)]$, where $\sum$ is the set of all $\mathcal{F}_k$-stopping times.

2. The time τ^* defined above is optimal: $\mathbb{E}[r^{-\tau^*}\hat{Y}(\tau^*, J_{\tau^*})] =\hat{Z}_{00}$.

Proof: see [2]

5.3.3 Change of measure

Letting $S_u, u<t$ be a price observed at an earlier date u, we know that it cannot be a martingale since it is risky when discounted by the risk-free rate. Therefore, under the probability measure $\mathbb{P}$ we cannot have

$$\mathbb{E}^{\mathbb{P}}[e^{-rt}S_t|S_u, u<t] =e^{-ru}S_u. \tag{147}$$

Instead, due to the existence of a risk premium we have

$$\mathbb{E}^{\mathbb{P}}[e^{-rt}S_t|S_u, u<t] > e^{-ru}S_u. \tag{148}$$

The key in finding an equivalent martingale measure $\widetilde{\mathbb{P}}$ such that the equation

$$\mathbb{E}^{\widetilde{\mathbb{P}}}[e^{-rt}S_t|S_u, u<t] = e^{-ru}S_u \tag{149}$$

is satisfied, is to switch the drift term to zero. This is done by switching the process W_t to a new process $\tilde{W}_t$ with distribution $\widetilde{\mathbb{P}}$. [1]

In order to find a probability measure $\widetilde{\mathbb{P}}$ such that S_t becomes a martingale we need to define a new probability $\widetilde{\mathbb{P}}$ by

$$Y_t \sim N(\widetilde{\mu}t, \sigma^2 t), \tag{150}$$

where the new drift parameter $\widetilde{\mu}_t$ defines the difference between the two measures $\mathbb{P}$ and $\widetilde{\mathbb{P}}$. We then have

$$\mathbb{E}^{\widetilde{\mathbb{P}}}[e^{-r(t-u)}S_t|S_u, u<t] = S_u e^{-r(t-u)} e^{\widetilde{\mu}(t-u)+\frac{1}{2}\sigma^2(t-u)}. \tag{151}$$

We define $\widetilde{\mu}$ as

$$\widetilde{\mu} = I - \frac{1}{2}\sigma^2, \tag{152}$$

where I defines the intranational inflation specified by the consumer price index CPI. When relating the inflation to commodity prices we have

$$I_t = r_t^N - F_t + \lambda_t^N \nu_t, \tag{153}$$

which is essentially a local expression of the so-called Fisher equation. It relates the change in fuel price to the nominal interest rate minus the implicit convenience yield F_t plus a risk premium term. [4]

By taking the inflation rate as the risk-neutral driver of the fuel price process, the fuel price process is supposed to behave like

$$\frac{d\tilde{S}_t}{\tilde{S}_t} = [r_t^N - F_t + \lambda_t^N \nu_t]dt + \nu_t d\tilde{W}_t. \tag{154}$$

The nominal risk-free rate r^N in equation (154) can be replaced by

$$r^N = I + F. \tag{155}$$

In a risk-neutral environment the risk premium term $\lambda_t^N \nu_t$ vanishes. Therefore we are left with

$$\frac{d\tilde{S}_t}{\tilde{S}_t} = I dt + \sigma d\tilde{W}_t. \tag{156}$$

When assuming a dynamic change in the the CPI we need a change-of-measure density process which switches the CPI to a risk-neutral world. This helps us to model the risk-neutral drift term of the fuel price process. Hughston showed that the change-of-measure density process ρ_t corresponds to the approach proposed above as

$$\rho_t = exp[-\int_0^t \lambda_t^N dW_s - \frac{1}{2}\int_0^t (\lambda_t^N)^2 ds], \tag{157}$$

where λ_t^N is the nominal risk premium vector.

Coming back to the determination of the risk-neutral probability measure for the fuel price process, we want the exponential on the right-hand side of (151) to become one. This is verified by (152) since for

$$-I(t-u) + \widetilde{\mu}(t-u) + \frac{1}{2}\sigma^2(t-u) = 0, \tag{158}$$

by substituting this in (151), we get

$$\mathbb{E}^{\widetilde{\mathbb{P}}}[e^{-I(t-u)} S_t | S_u, u<t] = S_u. \tag{159}$$

Transferring e^{Iu} to the right we get the equivalent martingale measure under $\widetilde{\mathbb{P}}$,

$$\mathbb{E}^{\widetilde{\mathbb{P}}}[e^{-It} S_t | S_u, u<t] = e^{-Iu} S_u. \tag{160}$$

By determining a particular value for $\widetilde{\mu}$ with the help of (154), we can find a probability distribution under which the expectation of the fuel price process fulfill the martingale property. This distribution is given by

$$N((I - \frac{1}{2}\sigma^2)t - F_t, \sigma^2 t). \tag{161}$$

5.4 Application

In the sections above we showed that from theoretical pricing point of view, we may deal with the continuous or the discrete case. The implementation of a fuel hedge in a discrete time framework is sufficient since it enables the consumer, who buys an American call option on fuel, to exercise his right every day when we consider daily time steps. By application of the binomial option pricing model we calculate following examples for the buyer of an American call option on fuel:

Fuel/Maturity(in months)	1	3	6	12
Diesel	0.01971436	0.03633417	0.05434838	0.08265938
Eurosuper	0.01865087	0.03449715	0.05177956	0.0791151
Normal	0.01871715	0.03461161	0.05193954	0.07933563
Super Plus	0.01466224	0.02762051	0.04219195	0.06596481

Figure 2: Call premiums (in Cent/Liter) of Diesel, Normal 90 octane, Super 95 octane and Eurosuper unleaded for 1, 3, 6 and 12 months maturity as of April 18th, 2012

Keeping the strike at-the-money we can see for different maturities how the price for hedging fuel develops. The price reflects the premium the buyer of such an option is obliged to pay if he wanted to buy his excerice right of physical delivery.

Considering the daily exercise possibility we face following chart for different hedging maturities:

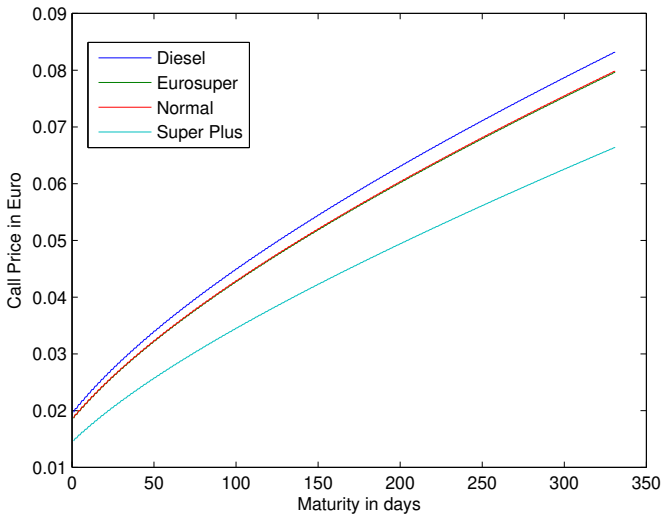

Figure 3: Call premium development (in Cent/Liter) of Diesel (blue), Eurosuper unleaded (green), Normal 90 octane (red) and Super Plus (light blue) for different maturities (1 day - 365 days) as of April 18th, 2012

In other words, the buyer of the fuel option has the right to buy 1 liter of fuel within the next year for todays price, if he payed a premium of e.g. 8 cents per liter. From pricing point of view it is possible to consider various strikes but in order to fulfill the requirements of a covered primary hedge, which is ought to prevent speculation, we propose just the implementation of at-the-money strike fuel options.

For the usual risk-neutral dynamics, under the assumption of a geometric brownian motion, we have

$$\frac{d\tilde{S}}{\tilde{S}} = (r-F)dt + \sigma d\tilde{W}, \tag{162}$$

where r displays the constant interest rate and F the convenience yield.[5] In order to get a Call price for a commodity option we can write

$$C(t) = e^{-r(T-t)}\mathbb{E}^{\tilde{P}}[max(0,S(T)-K)]. \tag{163}$$

Assuming that all commodities have the same expected return we may consider that for an end product, which has it's origin in a commodity, the inflation rate may serve as a valid risk-neutral measurement indicator. As equation (155) showed we might apply inflation as the only driver of the commodity economy. In order to see what difference in pricing this makes we compare a $T=$ 12-months American fuel Call option under the risk neutral measures $\tilde{P}_r$ and $\tilde{P}_I$. For calculation we make use of the CRR-model and evaluate

$$C(t) = e^{-r(T-t)}\mathbb{E}^{\tilde{P}_I}[max(0,S(T)-K)], \tag{164}$$

as well as

$$C(t) = e^{-r(T-t)}\mathbb{E}^{\tilde{P}_r}[max(0,S(T)-K)]. \tag{165}$$

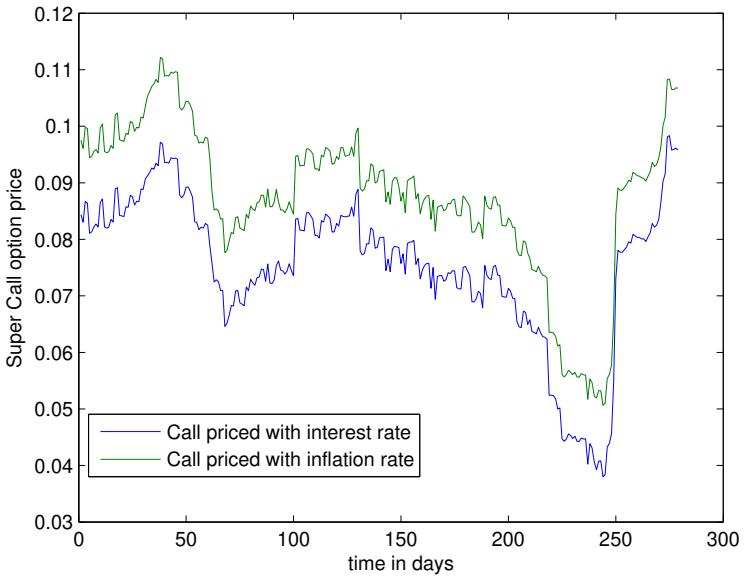

Figure 4: Daily Super Call premium development for fixed 12-month maturity for the period of Jan 10th 2011 - March 31st 2012.

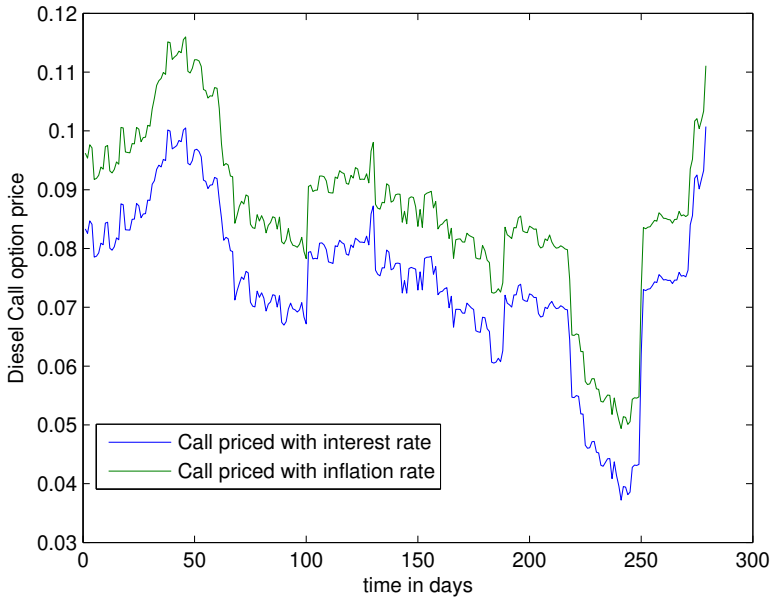

Figure 5: Daily Diesel Call premium development for fixed 12-month maturity for the period of Jan 10th 2011 - March 31st 2012.

We can see that the difference in the option premium comes from the difference in inflation and interest rate levels. Of course, the spread between the two pricing approaches tightens when looking at shorter maturites. Interpreting the spread difference is quite interesting since a higher Call option premium in an inflated environment means that hedging is more effective since future value depreciation is covered better by the fair value. In contrast to this a higher Call option premium in the interest rate world would mean that hedging is overpriced since future value depreciation is higher as the fair value might indicate.

5.5 Conclusion

In this paper we proposed a hedging approach for fuel. As fuel prices represent a key driver of inflation we show that by pricing of an American call option a fuel hedge for the prospective consumption within a certain time period is possible for paying a reasonable premium. In a risk-neutral environment this premium depends on the inflation drift which displays any increase in commodity prices in general. This in turn heavily influences the determination of the price of an option which is a derivative of an underlying such as crude oil. We showed two different frameworks in which such options could be priced, the continuous framework and the discrete one. For practical implementation the discrete time framework seems to be sufficient when considering daily time steps as consumption is assumed not to be necessary two times a day. Therefore the exercise times can be reduced to daily time intervals.

This procedure improves calculation time and provides a policy guideline.

For regulatory hedging reasons we just consider at-the-money American call fuel options which prevent speculation. Nevertheless the maturity may differ. The goal of fuel hedging is to reduce consumption costs by defining a worst case scenario over a certain time period.

5.6 Author contribution

The basic idea to develop a model for hedging of fuel prices came from S.U. K.A. proposed a binomial model after doing an extensive literature review. S.U. provided the data and K.A. performed the calculations and programming. S.U. & K.A. interpreted and summarized the results together and wrote the conclusion.

References

[1] Neftci Salih N., An Introduction to the Mathematics of Financial Derivatives, *Academic Press, Elsevier Science*, 1996, p.105.

[2] Davis H.A. Mark, American options in the binomial model (Revised), *Finite Difference Methods*, 2011.

[3] Lamberton Damien, Optimal stopping and American options, *Ljubljana Summer School on Financial Mathematics*, 2009.

[4] Hughston P. Lane, Inflation Derivatives, *Working paper*, 1998.

[5] Hélyette Geman, Commodities and Commodity Derivatives, *John Wiley& Sons Ltd*, 2005.

[6] Federal Ministry of Economy, Family and Youth (http://www.bmwfj.gv.at), *Fuel price monitor, Prices on energy, Energy and mining*, Jan 2013.