Corvinus University of Budapest
Faculty of Economics

Financial bubble detection and prediction

investigating the relationship

between the S&P 500 index absolute returns and the illiquidity parameter

Author: Péter Fürjész
Applied Economics
BSC thesis summary
2022

Contents

List of Figures

Abstract

In my thesis, I use Conic Finance to estimate the bid and ask prices of European index options based on real data. I calibrate the gamma parameter of the (Wang-distorted) Conic Black-Scholes model to real bid and ask prices between 2022-04-14 and 2022-11-16. For the calibration, I use the Black-Scholes implied volatility calculated from last prices as the volatility parameter. I calibrate the gamma parameter for twenty call and twenty put options, with four different window sizes, for a total of one hundred and sixty "simple" gamma time series.

From the simple gamma time series, I construct two different composite gamma time series. The aim is to produce time series that are representative of the trading as a whole. In the first case, for each window size, I consider only the gamma value of the option closest to the at-the-money level. As a result, I simplify the simple gamma time series into eight time series. I will refer to these time series as ATM gamma time series. In the second case, for each window size, I consider the gamma value calculated for all strikes, but inverted proportionally by the squared deviation from the at-the-money level. This means that the gamma value of ATM options has been given a higher weight in the market gamma calculation. In this case, I "compress" the simple gamma time series into eight time series. I refer to these time series as weighted gamma time series. I investigate the relationship between the lagged values of the above 16 time series and the current period absolute returns of the S&P 500 index using a VAR model.

Based on the VAR models, I reject my hypothesis that there is a significant negative relationship between the current period absolute return and the lagged values of the ATM gamma time series or the weighted gamma time series for any window size.

Key Words:

Conic Finance, Illiquidity parameter, Black -Scholes model, Wang-transformation

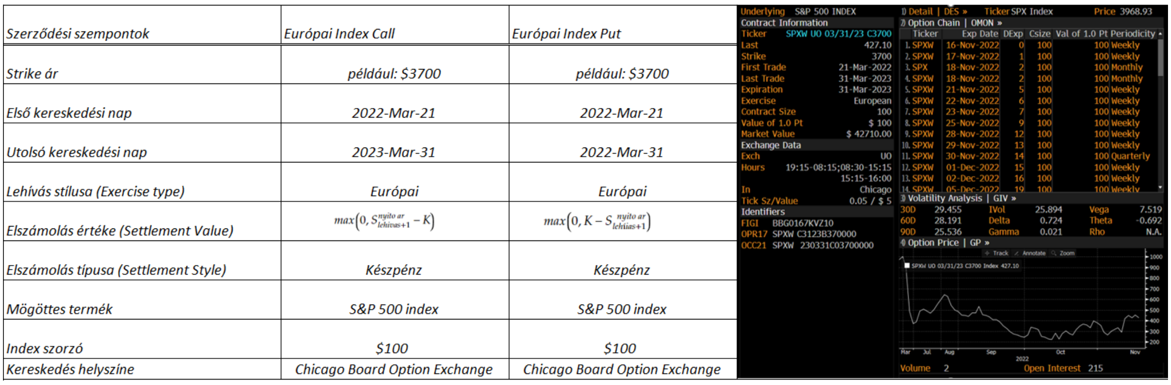
JEL classification: *G01, G12*

1 Introduction of the data used for the calculations

In my thesis, I examine European options contracts on the S&P 500 index issued by the CBOE, which are specified as follows:

Source: CBOE és Bloomberg

Figure 1: Specifications of the contract under investigation



Szerződési szempontok	*Európai Index Call*	*Európai Index Put*
Strike ár	*például: $3700*	*például: $3700*
Első kereskedési nap	*2022-Mar-21*	*2022-Mar-21*
Utolsó kereskedési nap	*2023-Mar-31*	*2022-Mar-31*
Lehívás stílusa (Exercise type)	*Európai*	*Európai*
Elszámolás értéke (Settlement Value)	$max\left(0, S^{nyito\ ar}_{lehivas+1} - K\right)$	$max\left(0, K - S^{nyito\ ar}_{lehíias+1}\right)$
Elszámolás típusa (Settlement Style)	*Készpénz*	*Készpénz*
Mögöttes termék	*S&P 500 index*	*S&P 500 index*
Index szorzó	*$100*	*$100*
Kereskedés helyszíne	*Chicago Board Option Exchange*	*Chicago Board Option Exchange*

Underlying S&P 500 INDEX

Contract Information

Ticker	SPXW UO 03/31/23 C3700
Last	427.10
Strike	3700
First Trade	21-Mar-2022
Last Trade	31-Mar-2023
Expiration	31-Mar-2023
Exercise	European
Contract Size	100
Value of 1.0 Pt	$ 100
Market Value	$ 42710.00

Exchange Data

Exch	UO
Hours	19:15-08:15;08:30-15:15 15:15-16:00
In	Chicago
Tick Sz/Value	0.05 / $ 5

Identifiers

FIGI	BBG0167KVZ10
OPR17	SPXW C3123B370000
OCC21	SPXW 230331C03700000

1) Detail | DES » Ticker SPX Index Price 3968.93

2) Option Chain | OMON »

	Ticker	Exp Date	DExp	Csize	Val of 1.0 Pt	Periodicity
1.	SPXW	16-Nov-2022	0	100	100	Weekly
2.	SPXW	17-Nov-2022	1	100	100	Weekly
3.	SPX	18-Nov-2022	2	100	100	Monthly
4.	SPXW	18-Nov-2022	2	100	100	Monthly
5.	SPXW	21-Nov-2022	5	100	100	Weekly
6.	SPXW	22-Nov-2022	6	100	100	Weekly
7.	SPXW	23-Nov-2022	7	100	100	Weekly
8.	SPXW	25-Nov-2022	9	100	100	Weekly
9.	SPXW	28-Nov-2022	12	100	100	Weekly
10.	SPXW	29-Nov-2022	13	100	100	Weekly
11.	SPXW	30-Nov-2022	14	100	100	Quarterly
12.	SPXW	01-Dec-2022	15	100	100	Weekly
13.	SPXW	02-Dec-2022	16	100	100	Weekly
14.	SPXW	05-Dec-2022	19	100	100	Weekly

3) Volatility Analysis | GIV »

30D	29.455	IVol	25.894	Vega	7.519
60D	28.191	Delta	0.724	Theta	-0.692
90D	25.536	Gamma	0.021	Rho	N.A.

4) Option Price | GP »

Volume 2 Open Interest 215

The specifications are based on data from the Bloomberg security description (DES) and the contract specifications section of the CBOE website [1]. The price of the contract is influenced by a number of other parameters, such as margin requirements, which in the case of exchange traded options are regulated by the exchange itself, in this case by the CBOE. These parameters, which are not introduced, are not part of any of the models presented below.

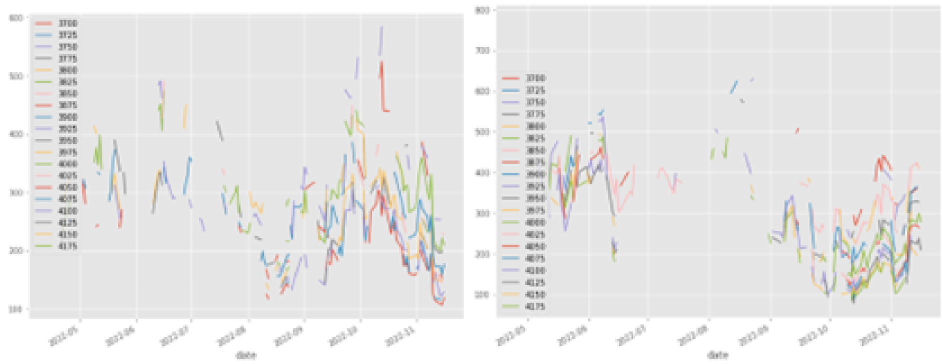
The data table contains the bid,ask and last prices of twenty put and call options with strike symmetrically positioned around (with respect to) the spot price of the underlying product at the time of the writing. This means that there was a 1-1 ratio of in-the-money and out-of-the-money put and call options at the time the option was written. The data table contains the bid and ask prices of put and call on almost every day, but last price, i.e. the price of the realized transactions, are only available on the 1/3 of the days. In the data table, observations are included from 2022-04-14 to 2022-11-16.

An important element of the subsequent calculations is the Black-Scholes implied volatility, which requires the last prices of the options under investigation. For this

[1] https://www.cboe.com/tradable_products/sp_500/spx_options/specifications/

Source: Own python visualization

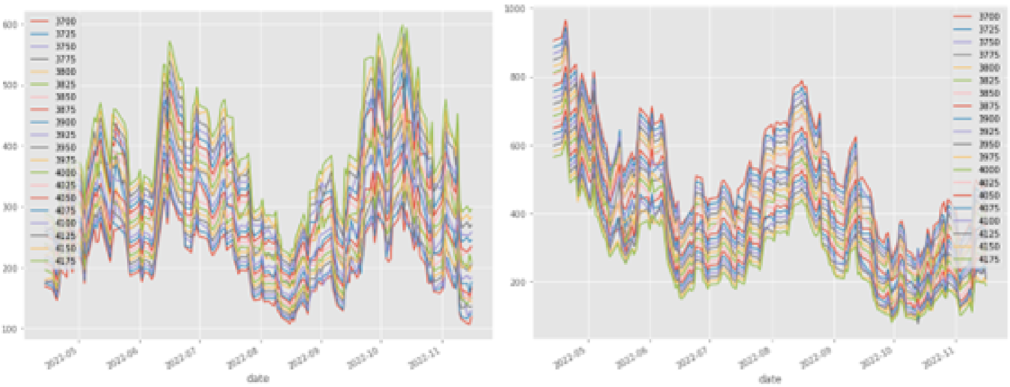
Figure 2: Last prices of Put (left) and Call (right) options before data substitution



purpose, the missing last prices were substituted with the arithmetic average (mid price) of the same day's bid and ask prices of the same contract. There was no time when bid and ask prices were not available.

Source: Own python visualization

Figure 3: Last prices of Put (left) and Call (right) options after data substitution



2 Conic Black-Scholes model

2.1 Black-Scholes model

Definition: Let $(\Omega, \mathcal{F}, P)$ be a probability space. Then $(W_t)_{t\geq 0}$ is a Brownian motion or Wiener process, if it satisfies the following conditions:

1. All elements of process $(W_t)_{t\geq 0}$ are independent and constant incremental, more precisely $W_y - W_t$ and $W_s - W_v$ are independent for all $v < s \leq t < y$.
2. $(W_t)_{t\geq 0}$ has continuous trajectory.
3. $W_t - W_s$ is normally distributed with zero mean and $\sqrt{t-s}$ standard deviation i.e. $W_t - W_s \sim N(0, \sqrt{t-s})$ for all $0 \leq s < t$.

Furthermore, let's call the Wiener process a Standard Brownian motion if its initial value is zero.
Assume that the returns of the underlying product follow a geometric Brownian motion, i.e. :

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) \tag{1}$$

Where μ is the drift term and σ is the volatility.

Let $S_0 = x_0$ then the solution of the stochastic differential equation is given by the Ito lemma:

$$S(t) = e^{\log x_{0+\hat{\mu}t+\sigma W(t)}} = S(0)e^{\hat{\mu}t+\sigma W(t)} \; ahol \; \hat{\mu} = \mu - \frac{1}{2}\sigma^2 \tag{2}$$

Then $S(t)$ is log-normally distributed, furthermore:

- $Var(S(t)) = S(0)^2 * e^{2\mu t}\left(e^{\sigma^2 t} - 1\right)$
- $E[S(t)] = S(0)e^{\mu t}$

(7) gives the following relation between two arbitrary times:

$$ln\left(\frac{S(t_2)}{S(t_1)}\right) = \hat{\mu}(t_2 - t_1) + \sigma(W(t_2) - W(t_1)) \tag{3}$$

By substituting $t_2 - t_1 = \Delta t \; and \; t_1 = t$ and using the property of the Wiener process that the change for time t is a random variable $N\left(0, \sqrt{t}\right)$. Furthermore:

- $\mu = r$ where r annual return on a risk-free deposit/bond
- S_t the value of the exchange rate at the start of the simulation
- σ volatility, the standard deviation of returns
- N standard normal random variable

Then:

$$S_t = S_{t-\Delta t} \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} z_t\right) \tag{4}$$

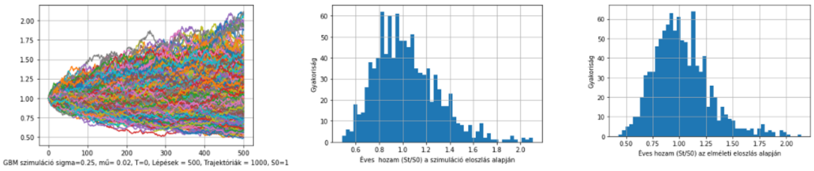
Based on equation (4), the GBM exchange rate model can be simulated, taking into account that $N\left(0, \sqrt{t}\right) = \sqrt{t} * N(0, 1)$ and z_t -s are independent standard normally distributed random variables. Let:

- $\sigma = 0.25$
- $\mu = r = 0.02$,
- The number of trajectories 1000
- The number of steps 500

Then, $t = \frac{1}{1000}$. Let's compare the distribution of the returns calculated for the whole period for each simulation (S_1/S_0) with the theoretical log-normal distribution. for this I use the function *numpy.random.lognormal* whose inputs are the parameters of the underlying normal distribution i.e.: expected value $\hat{\mu}T$, where $\widehat{\mu} = \mu - \frac{1}{2}\sigma^2$ and standard deviation $\sigma\sqrt{T} = \sigma * 1$. From this distribution I generate a number of pieces equal to the number of trajectories, using $log(S_{t+1}) - log(S_t) = log\left(\frac{S_{t+1}}{S_t}\right)$. Then the results of the simulation:

Source: Own python simulation

Figure 4: Geometric Brownian motion - simulation

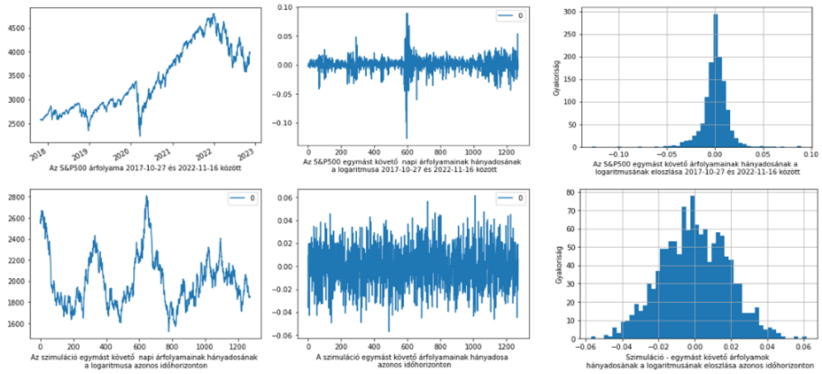


Let's use the Fisher-Pearson coefficient to measure the skewness, $g_1 = \dfrac{m_3}{m_2^{3/2}}$ where $m_i = \dfrac{1}{N}\sum\limits_{n=1}^{N}(x[n] - \overline{x})^i$. Negative values of skewness indicate left-skewed data, and positive values indicate right-skewed data. Left skew means that the left tail is "long" compared to the right tail. Similarly, right skew means that the right tail is "long" compared to the left tail. If the data are multimodal, this may affect the sign of skewness. The Fisher-Pearson coefficient for the simulated data 0.792 while for values generated from a theoretical distribution 0.974. Both results are consistent with the assumption that the distribution is logarithmic, since the log-normal distributions are positively (right) skewed with heavy (long) right tails, due to low mean values and large variances of the random variables.

Let quantify the kurtosis as the standardised fourth central moment, i.e. $\dfrac{\sum_{i=1}^{N}(Y_i - \overline{Y})^4/N}{\sigma^4}$. If we use the Fisher definition, we have to subtract three from the result to get zero for a normal distribution. For negative values, the distribution is flatter than normal, for positive values it is more peaked. The Fisher indicator for the simulation is 0.905, while for the values generated from the theoretical distribution it is 1.305, so in both cases we can talk about a distribution that is more peaked than the normal. Let us compare the distribution of the logarithms of the daily percentage returns with the distribution of the logarithms of the percentage returns within a trajectory.

Source: Own python simulation

Figure 5: GBM simulation vs. real data



In the previous case (Figure 4), the ratio of the initial values to the final value was calculated on the basis of five hundred simulations, and the latter value was "generated" five hundred times from the corresponding log-normal distribution.

Conversely, the distribution of the logarithms of the percentage returns within a trajectory should be close to normal. (Figure 5 bottom row) If our assumption that real exchange rates also follow a GBM (Geometry Brownian motion) is true, then we also obtain a close to normal distribution for real values. (Figure 5 top row) In both cases we have 1273 observations. The Fisher-Pearson coefficient is -0.017 for the simulated data and -0.825 for the real data. This means that the simulated data follow a nearly symmetric distribution, while the real daily observations follow a slightly right skewed distribution in the sample. The Fisher's peak value for the simulation was 0.068, while for the real data 13.534. This means that the peak of the simulated data is close to the peak of the normal distribution. In contrast, the distribution of the real data is significantly more peaked, i.e. small fluctuations are much more frequent than would be expected from the GBM. The slight skewness and significant peakedness of the real data is commonly referred to as the *asymmetric leptokurtic* property.

The F. Black 1973 option pricing model *hedge* has been a major breakthrough in the pricing of European options. The model has had and continues to have a major impact on the way traders price and hedge (*hedge*) European options. The model makes a number of assumptions about the underlying product, the market and the option contract.

1. Short-selling is allowed
2. No taxes and transaction costs and the underlying product is perfectly dividable
3. There are no arbitrage opportunities on the market
4. The trading is continuous
5. The interest rate is constant and the same for all maturities
6. Assumes that the underlying product price follows a Geometric Brownian Motionv where μ and σ fixed,i.e.:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) \tag{5}$$

Denote the price of an option $f = f(t,\ (S(t))$, meaning that the option price (in addition to the constant strike price and the risk-free rate of return) depends only on time and the price of the underlying. Using the Ito-lemma:

$$df = \left(\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)dt + \frac{\partial f}{\partial S}\sigma S dW \tag{6}$$

Suppose you are in a short position in a particular option. We would like to add a sufficient amount of shares to our portfolio so that the value of the portfolio (currently unknown) does not change due to a change in the Wiener process (over time dt). The value of the option depends on the Winer process in the interval dt only according to $\frac{\partial f}{\partial S}\sigma S dW$. We would like to neutralize this "effect" by holding a sufficient amount of the underlying product.

Consider the following portfolio: take a short position in an options contract and a short or long position of size $\partial f/\partial S$ in the underlying product. Denote the value of the portfolio by X(t) and then its change according to (6):

$$dX = d\left(\frac{\partial f}{\partial S} * S - f\right) = \frac{\partial f}{\partial S} * dS - df = \tag{7}$$

$$= \frac{\partial f}{\partial S} * (\mu S dt + \sigma S dW) - \left(\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)dt - \frac{\partial f}{\partial S}\sigma S dW \tag{8}$$

Then it can be seen that the terms $\frac{\partial f}{\partial S}\mu S * dt$ and the $\frac{\partial f}{\partial S}\sigma S dW$ are eliminated. Thus, the dependence on the Wiener process is eliminated in the time interval dt.

$$dX = -\left(\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)dt \tag{9}$$

Since (9) no longer depends on the Wiener-process, which is the only risk in the model, the change in portfolio value should be equal to the increase in the risk-free return on the portfolio value over dt.

$$-\left(\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)dt = \ rXdt \tag{10}$$

Substituting X:

$$-\left(\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)dt = \ r\left(\frac{\partial f}{\partial S} * S - f\right)dt \tag{11}$$

Rearranging the equation:

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \tag{12}$$

The (12) equation is the Black-Scholes partial differential equation. By solving this differential equation and using the boundary condition $f = \max(S - K, 0)$ when $t = T$ for call or $f - \max(K - S, 0)$ when $t - T$ for put, the solution of the (non-stochastic) partial differential equation for call option pricing is well known:

$$c = S_0 N(d_1) - Ke^{-rT} N(d_2) \tag{13}$$

$$p = Ke^{-rT} N(-d_2) - S_0 N(-d_1) \tag{14}$$

$$ahol:$$

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)\,T}{\sigma\sqrt{T}} \tag{15}$$

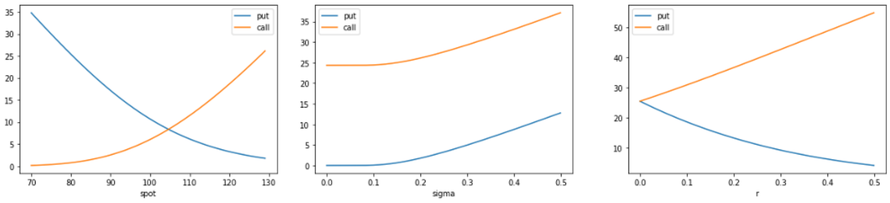
$$valamint:$$

$$d_2 = \frac{\ln(S_0/K) + (r - \sigma^2/2)\,T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \tag{16}$$

It is interesting to examining how the price of a theoretical option contract $(K = 100, T = 1)$ evolves if the other non-contractual parameters are changed:

Source: Own python calculation

Figure 6: Sensitivity of the Black-Scholes model to non-contractual parameters



As the spot price changes, the value of the call option (leftmost graph) increases non-linearly, while the value of the put option decreases non-linearly. This is expected, since the derivative of equation (13) by S_0 is positive for call and negative for put.

Conversely, an increase in volatility (middle graph)) increases the price of both put and call options. This is to be expected, since if volatility increases, the probability that the option will have a positive payoff increases and in addition $\frac{\delta c}{\delta \sigma} > 0$ and $\frac{\delta p}{\delta \sigma} > 0$.

The effect of r is also reasonable (rightmost graph), as in a favourable interest rate environment, investors are willing to pay more for a call option, as they can keep the share price in deposit at a higher yield. In the case of a put option, the obligor can invest the premium at a higher rate, furthermore: $\frac{\delta c}{\delta r} > 0$ and $\frac{\delta p}{\delta r} < 0$.

The only parameter of the Black-Scholes-Merton pricing formula that is not observable directly in the market is the volatility of the stock price. Suppose that the price of a call option is given, i.e. observable in the market. Denote the price by C^*. Then let us denote the implied volatility by σ^{imp}, which is the solution of the equation $C\left(S_t, K, t, T, r, \sigma^{imp}\right) = C^*$. In practice, the determination of σ^{imp} is called the calibration of the Black-Scholes model. This means solving the following univariate minimization problem (S_t, K, t, T, r, C^* is known) using some minimization algorithm:

$$\begin{aligned} min: \quad f(\sigma) &= C(\sigma) - C^* \\ if: \ 0.001 < \sigma < &\ 3 \end{aligned} \tag{17}$$

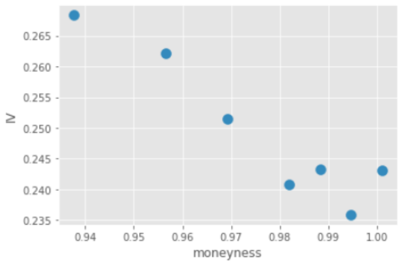
For the implied volatility calculation I used the *scipy.optimize.minimize_scalar* function and the optimization algorithm was set to bounded. In this case, the condition of equation (17) must be specified, more precisely, the interval at which the optimal value of sigma is sought using Brent's algorithm

The volatility smile is the geometric pattern of implied volatilities for a series of options with identical expiration dates. These implied volatilities, when plotted against strike prices, form a convex curve, hence the term "smile". Volatility smiles should never occur under standard Black-Scholes option theory, which usually requires a completely flat volatility curve.

The first observable volatility smile appeared after the 1987 stock market crash. In practice, this means that out-of-the-money options are traded at prices above the price justified by the Black-Scholes model.

Source: Own python calculation Appendix B / 9.0.4

Figure 7: Volatility smile 2022-09-04 - out-of-money calls



In Figure 7. volatility is plotted against moneyness, which is calculated by the formula $\frac{S_t}{K}$.

- An option is called an in-the-money (ITM) call (put) option if the current price of the underlying divided by the strike price of the option is less (greater) than one.
- An option is called an at-the-money (ATM) call option if the above ratio is close to one.
- An option is called an out-of-the-money (OTM) call (put) option if the above ratio is greater (less) than one.

At the time the option was written, the price of the S&P 500 index was 3946.01, while at the time of the calculation it was 4392.59. This means that the initial 1-1 ratio of in-the-money to out-of-the-money options has changed. The figure shows the negative slope segment of the volatility smile.

2.2 Application of the Black-Scholes model to estimate bid and ask prices

The Black-Scholes model (as traditional pricing theories in general) is based on the *law of one price*, while ignoring the effect of *market liquidity* on bid-ask spreads. In models based on the assumption of a price (or *equilibrium price*), we assume that we can buy and sell at the same price. In the conventional "world" represented by a risk-neutral measure, the price (or fair value) of a derivative is the (discounted) expected value under the risk-neutral measure, more precisely $V(X) = \exp(-rT)E_Q[X]$, where Q is the risk-neutral measure (*risk neutral measure*).

In contrast, in real markets, we can observe two prices, namely the price at which the market is willing to buy (bid) and the price at which the market is willing to sell (ask). The theory of conic finance replaces the *law of one price* with the *law of two prices* allowing market participants to sell to the market at the *bid price* and buy from the market at the higher *ask price*

The basis of conic finance is provided by the coherent risk measures presented by Artzner P. Artzner 1999. Kusuoka Kusuoka 2001 showed that there always exists a set of distortion functions Ψ_γ which are suitable as a "substitute" for the set of measures $\mathcal{D}_\gamma$ introduced later. Wang Wang 2002 introduced the Wang transformation as a distortion function, which later became the basis for many researches. The development of concic finance is associated with Madan and Cherry. Their publications A. S. Cherny 2006 A. S. Cherny 2008 A. S. Cherny 2010 are summarized in Madan's 2016 book D. Madan 2016.

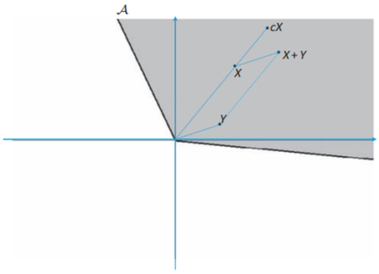
Several research papers have been written in the past years, mainly aimed to explain various derivative market phenomena and to determine the bid and ask prices of financial derivatives, based on the Conic Finance assumptions.

Karimov 2017 investigated the pricing of European index options by transforming the Black - Scholes model and the Kou model with conic finance. Z. Lia 2019 modified the Herston model in his research using the MINMAXVAR distortion function. W. Wang 2022 determines the bid and ask price of American options. M. Michielon and Spreij 2021 examined how the Black-Scholes implied volatility changes when last prices are replaced by observed bid and ask prices. The calculations are performed using a conic (Wang distorted) Black-Scholes model.

Consider a probability space $(\Omega, \mathcal{F}, P)$. Let's represent the payoff of a derivative at time T by a random variable variable X and let's call this a *risk* in the following. Let $\mathcal{A}$ be a convex set of these risks, and call it a convex set if for any $X, Y \in \mathcal{A}$ $0 \leq \alpha \leq 1, \alpha X + (1-\alpha)Y \in \mathcal{A}$.

Source: D. Madan 2016

Figure 8: Set of acceptable zero-cost cash flows



Furthermore, let's call $\mathcal{A}$ a cone if for any $X \in \mathcal{A}$, $c * X \in \mathcal{A}$ where $c > 0$. A risk measure $\rho\ (X)$ is a function that assigns a non-negative real number to a risk. Let $\rho\ (X)$ be a coherent risk measure that assigns the values of the $[0, \infty]$ interval to each random variable. Based on D. Madan 2016, the coherent measures assign values to the risks while satisfying the following four criteria:

1. $\rho(X + c) = \rho(X) + c$ (transitivity)
2. $\rho(X + Y) \leq \rho(X) + \rho(Y)$ (sub-additivity)
3. For any $c > 0$, $\rho(cX) = c\rho(X)$ (positive homogeneity)
4. If $P(X \leq Y) = 1$, then $\rho(X) \leq \rho(Y)$ (monotony)

where $X, Y \in \mathcal{A}$ and X and Y are non-negative probability variables:
Let X be an acceptable risk at γ level if $\rho\ (X)\ > \gamma$.

Then, based on A. S. Cherny 2010, there exists a set of probability measures $\mathcal{D}_\gamma$ such that:

$$\rho(X) \geq \gamma \iff E^Q[X] \geq 0 \text{ for any } Q \in \mathcal{D}_\gamma \tag{18}$$

Consider a scenario where γ is given. When someone is willing to sell a risk? If the expected value of the cash-flow margin from the sale under any $Q \in \mathcal{D}_\gamma$ measure, is positive. Let the price *ask* be the minimum price at which the market is willing to sell a risk X. Then for a given level of γ :

$$\begin{aligned} ask_\gamma(X) &= \inf\{\ a : \alpha(a - X) \geq \gamma\} \\ &= \inf\left\{a : E^Q[a - X] \geq 0 \text{ for any } Q \in \mathcal{D}_\gamma\right\} \\ &= \sup_{Q\in\mathcal{D}_\gamma} E^Q[X]. \end{aligned} \tag{19}$$

One can see that the seller is only willing to sell the uncertain cash flow at the "upper limit" of his valuations. Let *bid* be the maximum price the market is willing to pay for a risk X. Then for a given γ level:

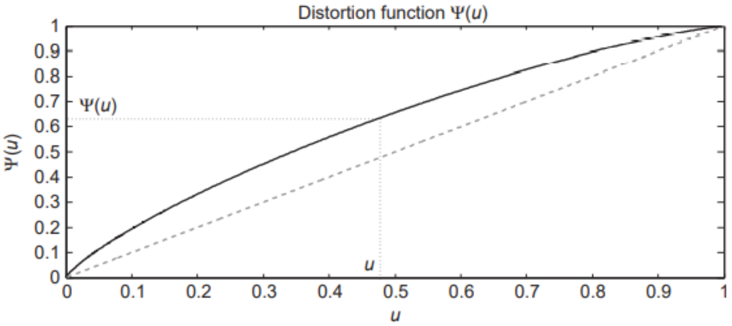
$$bid_\gamma(X) = \sup\left\{b : E^Q[X - b] \geq 0 \text{ for any } Q \in \mathcal{D}_\gamma\right\} = \inf_{Q\in\mathcal{D}_\gamma} E^Q[X]$$

One can see that, in the case of a random cash flow purchase, the buyer is only willing to pay the minimum of his "valuations" at given γ level.

Let's call the function $\Psi : [0,1] \to [0,1]$ a *distortion function* if and only if it is monotone and $\Psi(0) = 0$,$\Psi(1) = 1$. This means that, a concave distortion function is no more than a concave distribution function on the $[0,1]$ interval.

Source: D. Madan 2016

Figure 9: Simple distortion function



Let $\hat{\Psi}$ be the complementary distortion function of the Ψ distortion function, if $\hat{\Psi}(x) = 1 - \Psi(1 - x),\ \ x \in [0,1]$.

Assume that the random variable $F_X(x)$ follows a distribution F_X and the riskiness of a payout is determined only by the random variable's distribution. Under the hypotheses of co-monotone additivity and a dependence on just the distribution function results of Kusuoka 2001 imply that the bid and ask price must be an expectation under a concave distortion. More specially, there must exist a concave distribution (i.e. a distortion) from the unit interval to itself such that for any risk X with distribution function $F_X(x)$, we have:

$$\text{bid}(X) = \exp(-rT) \int_{-\infty}^{+\infty} x \mathrm{d}\Psi(F_X(x)) \tag{20}$$

and:

$$\text{ask}(X) = -\exp(-rT) \int_{-\infty}^{+\infty} x \mathrm{d}\Psi(F_{-X}(x)) \tag{21}$$

The distortion function has so far only been used as an abstract term. D. Madan 2016 intorduced five different distortion functions [2]. Consider the Wang trasform Wang 2002, i.e:

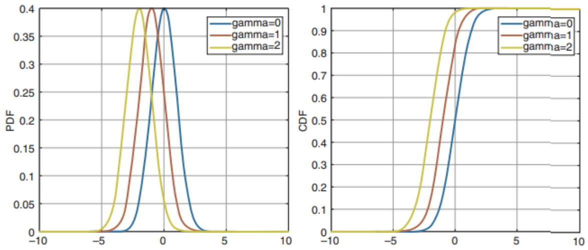
$$\Psi^{\gamma}(u) = \Phi\left(\Phi^{-1}(u) + \gamma\right), u \in [0,1], \gamma \geq 0. \tag{22}$$

distortion function which is a concave function, i.e. it assigns a higher probability to small values for larger gamma. Let us examine how the expected value of the normal distribution is modified by the distortion:

$$\begin{gathered} \Psi_{\gamma}^{\text{WANG}}\left(\mathrm{N}\left(\frac{x-\mu}{\sigma}\right)\right) = \mathrm{N}\left(\mathrm{N}^{[-1]}\left(\mathrm{N}\left(\frac{x-\mu}{\sigma}\right)\right) + \gamma\right) \quad = \\ \mathrm{N}\left(\frac{x-\mu}{\sigma} + \gamma\right) = \mathrm{N}\left(\frac{x-(\mu-\gamma\sigma)}{\sigma}\right) \end{gathered} \tag{23}$$

Source: Karimov 2017

Figure 10: Distortion of normal distribution by Wang transformation



It can be seen that the higher the γ, the higher the distortion. This means that the distribution function (23) assigns a higher probability to negative values than the original distribution function.

[2] These: MINVAR, MAXVAR, MAXMINVAR, MINMAXVAR and WANG-transform

Apply the distortion function to the Black-Sholes model. Denote the distribution of the option payoff by F_{CT}. Let $b_\gamma(C)$ be the bid price of a call option in case of "level" γ distortion.

Then, using equation (20) (without discounting):

$$b_\gamma(C) = \int_0^\infty (x - K) d\Psi^\gamma(F_{C_T}(x)) \tag{24}$$

The call option payoff can be calculated from the underlying product price S_T (stock price at time T) and the strike price . The option is only exercised above a stock price of K, so change the integration limits and integrate by S_t:

$$b_\gamma(C) = \int_K^\infty (x) d\Psi^\gamma(F_{S_T}(x)) \tag{25}$$

As we have previously seen, $S_T - S_{T-t}$ follows a lognormal distribution in the case of GBM.

Furthermore, the logarithm of the returns follows a normal distribution with expected value $\mu = \mu^* - \dfrac{1}{2}\sigma^{*2}$ and standard deviation $\sqrt{\Delta t} {*} \sigma^*$. Denote the exchange rate at t by S_t. Let T be the time of expiry. Then the distribution of the logarithm of the exchange rate at time T is normal, with $\mu = lnS_t + \left(\mu^* - \dfrac{1}{2}\sigma^{*2}\right)(T - t)$ expected value and with $\sigma = \sqrt{T-t} {*} \sigma^*$ standard deviation, where σ^* s μ^* are the parameters of the GBM. Let $\mu^* = r$. Then, $F_{lnS_T}(x)$ can be represented in normalized form: $\Phi\left[\dfrac{y - \ln S_t - \left(r - \frac{1}{2}\sigma^{*2}\right)(T-t)}{\sigma^*\sqrt{T-t}}\right]$. We have seen in equation (23) that the Wang distortion function reduces the expected value of the distribution by $\gamma * \sigma$. Then $\Psi^\gamma(F_{lnS_T}(x))= \Phi\left[\dfrac{y - \ln S_t - \left(r - \frac{1}{2}\sigma{*}^2\right)(T-t) + \gamma\sigma^*\ \sqrt{T-t}}{\sigma^*\sqrt{T-t}}\right]$ whereas $\sigma = \sigma^* * \sqrt{T-t}$. Substitute into equation (25):

$$b_\gamma(C) = \int_K^\infty x d\Phi\left[\frac{y - \ln S_t - \left(r - \frac{1}{2}\sigma{*}^2\right)(T-t) + \gamma\sigma^*\sqrt{T-t}}{\sigma^*\sqrt{T-t}}\right] \tag{26}$$

Karimov 2017 Appendix B solves the integral (26) and determine the formula for the bid and ask prices of the call and put options.

Option	Pricing formula	d_1	d_2
$b_\gamma(C)$	$S_l e^{-\gamma\sigma\sqrt{T-1}}\Phi(d_1) - e^{-r(T-t)}K\Phi(d_2)$	$\frac{\ln\frac{S_l}{K}+\left(r+\frac{1}{2}\sigma^2\right)(T-t)-\gamma\sigma\sqrt{T-I}}{\sigma\sqrt{T-t}}$	$d_1 - \sigma\sqrt{T-t}$
$a_\gamma(C)$	$S_l e^{\gamma\sigma\sqrt{T-l}}\Phi(d_1) - e^{-r(T-t)}K\Phi(d_2)$	$\frac{\ln\frac{S_l}{K}-\left(r+\frac{1}{2}\sigma^2\right)(T-t)+\gamma\sigma\sqrt{T-l}}{\sigma\sqrt{T-t}}$	$d_1 - \sigma\sqrt{T-t}$
$b_\gamma(P)$	$e^{-r(T-t)}K\Phi(d_2) - S_l e^{\gamma\sigma\sqrt{T-l}}\Phi(d_1)$	$\frac{\ln\frac{K}{S_l}-\left(r+\frac{1}{2}\sigma^2\right)(T-t)-\gamma\sigma\sqrt{T-t}}{\sigma\sqrt{T-t}}$	$d_1 + \sigma\sqrt{T-t}$
$a_\gamma(P)$	$e^{-r(T-t)}K\Phi(d_2) - S_l e^{-\gamma\sigma\sqrt{T-l}}\Phi(d_1)$	$\frac{\ln\frac{K}{S_l}-\left(r+\frac{1}{2}\sigma^2\right)(T-t)+\gamma\sigma\sqrt{T-t}}{\sigma\sqrt{T-t}}$	$d_1 + \sigma\sqrt{T-t}$

In my further calculations I will apply the formulas above [Appendix B / 7.0.3]

3 Investigating the relationship between the illiquidity parameter and the SP500 index

In the previous chapter, we saw that the γ parameter (via the distortion function) allows us to incorporate the market's expectation about the underlying product in the pricing. Based on what we have seen so far, one can see that changed expectations affect the liquidity of options trading and, through this, the price of particular options.

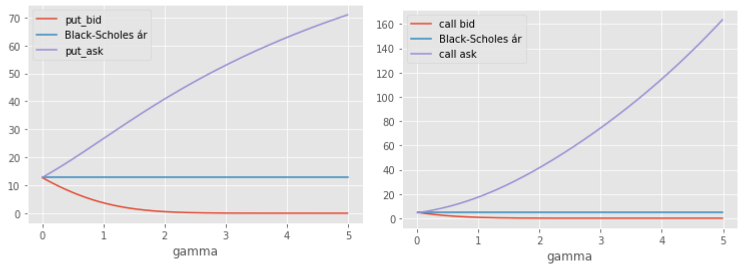
From (23), we can conclude that an increase in the gamma parameter "decreases" (in the case of the Black-Scholes model and the Wang transformation) the expected price of the underlying, but we have not seen how exactly it affects the price.

3.1 Estimation of the gamma parameter on real data

Consider an out-of-the-money call and an in-the-money put option. Let $K = 110,\ S_0 = 100, \sigma = 0.2 \geqq 0.02$.

Source: Own python calculations

Figure 11: Effect of increasing gamma on bid and ask prices



In the figure, the blue line with zero slope is the Black-Scholes price (which is not affected by the parameter γ). One can see that for $\gamma = 0$, the BS price is equal to the bid and ask prices (in this case the distortion is equal to zero).

Not clearly visible on the figure, but the BS price of the call option is 4.943 while the BS price of the put option is 12.765. We expected the call price to be lower as we are considering out-of-the-money call and in-the-money put option. On the other hand, the prices satisfy the put-call parity, i.e. $BS(call) - BS(put) = S_0 - K * e^{-rt}$, since: 4.943 $- 12.765 = -7.821$ and $100 - 110 * e^{-0.02*1} = -7.821$.

It can be seen that as the gamma parameter increases, the bid price decreases in both cases and the ask price increases in both cases, i.e. the spread increases. Negative gamma value is not possible, as in this case the bid price would be higher than the ask price.

How can the model be calibrated? In the case of the BS model, we have seen that the model was calibrated by choosing the value of the only unobservable parameter σ, which makes the BS price equal to the market price.

The conic or distorted BS model has two unobservable parameters, σ and γ. In my thesis, I approximate the σ parameter with the BS implied volatility.

The gamma parameter is then the only unobservable parameter to approximate. We have seen that for the simple BS model, when the BS implied volatility is substituted into the BS equation (if we calibrate to exactly one option price), we get back exactly the market price. In the case of conic BS, we cannot calibrate the model exactly to the bid and ask prices, taking into account we try to fit the model to two prices by optimizing a single parameter. Let approximate the parameter γ accordingly:

$$TSE_{bid,ask}(\gamma) = \sum_{i=1}^{\tau} \left((bid_i - b_{\gamma,i})^2 + (ask_i - a_{\gamma,i})^2 \right) \tag{27}$$

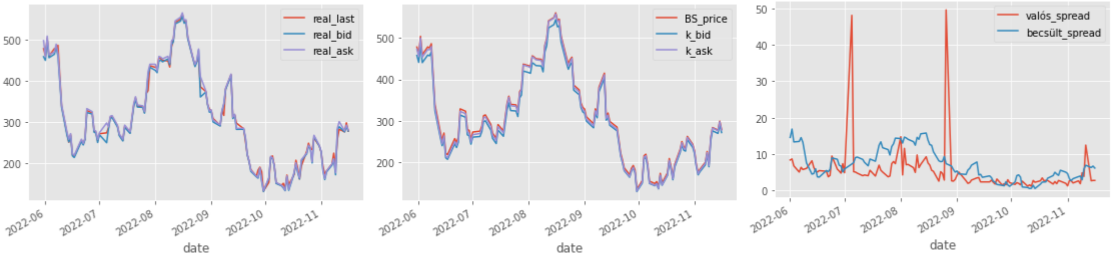
$$\begin{aligned} \min_{\gamma}: &\quad TSE_{\text{bid,ask}}(\gamma) \\ if: &\quad \gamma \geq 0 \end{aligned} \tag{28}$$

where τ (27) denotes the number of observations for which the model is optimized. The question arises whether to optimise for observations before or after the current period. Since the primary goal is to identify the bubble state in the real time, the optimization in my paper is always for previous observations (bid-ask prices).

In my thesis, I consider the interest rate to be constant: $r = 0.02$. The input data consists of the historical last, bid and ask prices of twenty put and twenty call contracts (with the same maturity 2023-03-31) and the last prices of the S&P500 index at the same time horizon. The option prices are available between 2022-04-14 and 2022-11-16, giving a total of 18150 prices. Select an (at the date of the writing) in-the-money call option, and let $K = 000$, $\tau = 14$.

Source: Own python calculations

Figure 12: Real vs. Estimated bid and ask prices



The leftmost graph shows the evolution of the real bid, last and ask prices over time. The middle plot shows the evolution over time of the bid and ask prices estimated using Conic BS and the last prices estimated using un BS. The BS last prices and the real last prices are the identical, because in the calculation $\sigma_t = \sigma_t^{imp}$. The rightmost figure shows the evolution of the real and the estimated spread over time. No extreme deviation is visible - the MAPE[3] value 38.104 - , but two outliers are observable in the Bloomberg data, which are data errors.

It is useful to clarify the meaning of gamma when using the Wang transformation. The gamma parameter can be interpreted as the market's implied expectation about the underlying product. The question arises as to which contract's gamma parameter should be taken into account. It is empirically observed that trading of the at-the-money options is the most active. Therefore it can be expected, that expectations are most likely to be reflected in the at-the-money options price.

Let's call at-the-money gamma time series the time series whose value at a given point in time is equal to the gamma value of the most at-the-money option.

Let's call a weighted gamma time series a time series whose value at a given

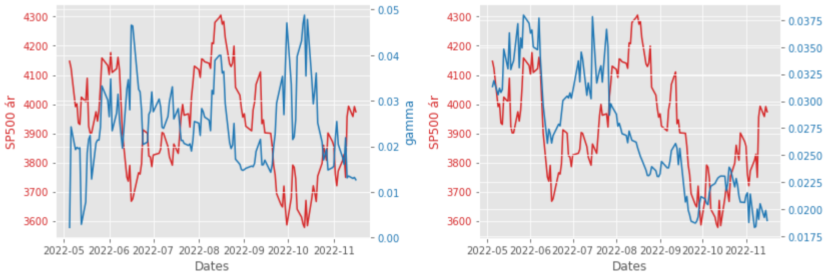
[3] $MAPE = \frac{1}{n} \times \sum \left| \frac{\text{real value } - \text{ estimated value}}{realvalue} \right|$

point in time is equal to the total weighted sum of the gamma parameters of each option weighted in inverse proportion to its at-the-moneyness. More precisely:

Let's $\tau = 7$ and take a put or call option with K strike. Let $\gamma_K(t)$ denote the option's approximated $\hat{\gamma}t$ parameter at time t furthermore, let $w_K(t)$ be the weight of $\hat{\gamma}_K(t)$ at time t, where $w_K(t) = \left(1-(1-M_K(t))^2\right)$ and $M_K(t)$ denotes the option's moneyness at time t . Then the value of the weighted gamma (WG) series at time t is $WG(t) = \sum_{K\in\ elrhet\ K} w_K(t) * \gamma_K(t)$.

Source: Own python calculations

Figure 13: ATM gamma and weighted gamma time series of call options with $\tau = 14$ parameter setting



The figure figure on the right shows the weighted gamma time series of call options and the figure on the left shows the at-the-money gamma time series of call options both calculated with $\tau = 14$ parameter setting.

3.2 Result of the VAR model

I calculate both at-the-money gammas and weighted gammas with $\tau \in [1, 7, 14, 28]$ parameters. I perform these calculations for both call and put options. Thus, I examine the relationship between a total of 16 time series and the absolute returns of the SP 500 index using a VAR model. The code and data used for the statistical calculations are available at https://github.com/pfurjesz/Szakdolgozat_2022 under the name: **VAR Models**.

3.2.1 At-the-money gamma time series

Source: Own python calculations

Figure 14: Evolution of ATM gamma values for different tau parameters

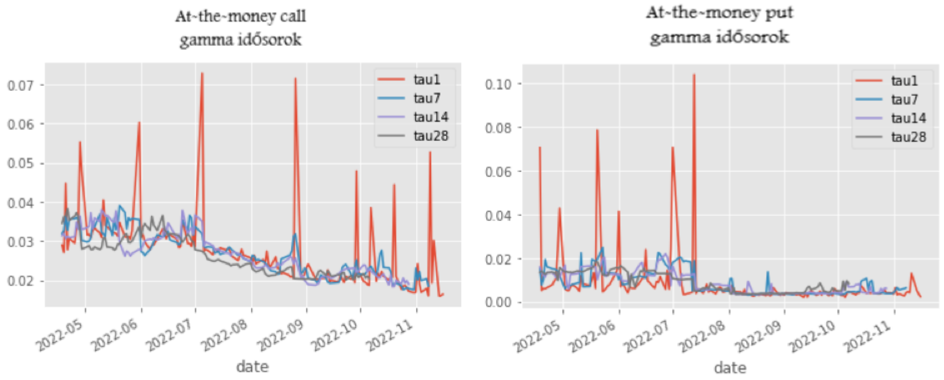


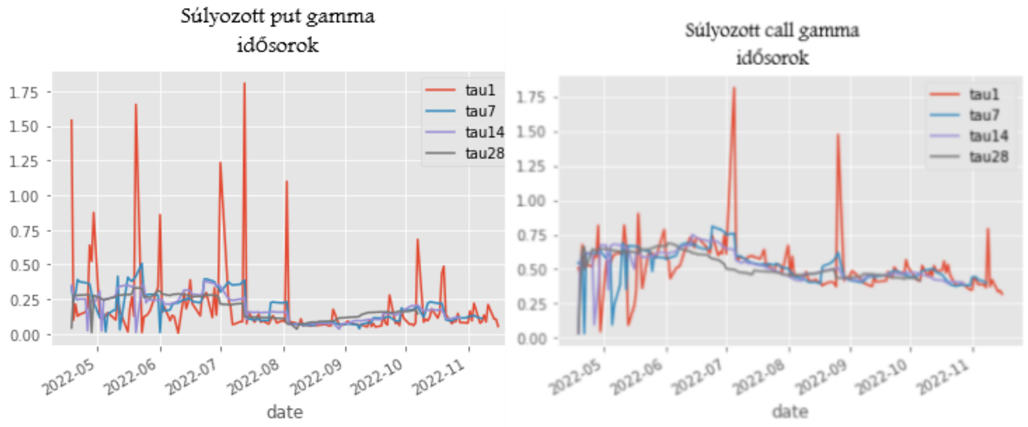
Figure 14 shows that for $\tau = 1$ the gamma time series is very sensitive to outliers, since in this case it is calibrated for only two observations. The VAR models examining the relationship between the different ATM gamma time series and the absolute returns of the SP 500 index are summarized in Appendix C 9.0.1 and 9.0.2. None of the models explaining $\Delta S\&P500$ were significant.

The models explaining the value of gamma were significant in several cases. In all of these models, the significant variable was the lagged gamma, and therefore the gamma time series in these cases can be said to be autoregressive, since at a significance level of $\alpha = 1\%$, the lagged gamma values have a significant effect on the current period gamma value. The adjusted R^2 was below 30% in these cases as well. For these models both the p value of the Portmanteau-test and the p value of the Breusch-Godfrey test were below 0.05, so that the error terms cannot be considered as white noise either together or separately.

3.2.2 Weighted gamma time series

Source: Own Python calculation

Figure 15: Evolution of weighted gamma values for different tau parameters



For the weighted gamma time series (Figure 15), it can also be observed that for $\tau = 1$ the gamma time series is very sensitive to outliers. VAR models examining the relationship between different weighted gamma time series and the absolute returns of the SP 500 index are summarized in Appendix C 9.0.3 and 9.0.4. None of the models explaining the $\Delta S\&P500$ were significant. The models explaining the value of gamma were significant in several cases and in these cases the gamma time series were also autoregressive. For these models, both 0ortmanteau test's and the Breusch-Godfrey test's p value were below 0.05, so that the error terms cannot be considered as white noise either jointly or separately.

4 Summary

In my thesis, I investigated the relationship between the illiquidity parameter approximated from the bid and ask prices of the European S&P 500 index option and the absolute returns of the S&P500. I presented the Black-Scholes model and the Geometric Brownian motion used to model the underlying product. I showed that the real returns follow a different distribution than assumed by the GBM.

Conic Finance resolves the assumption of one price with the assumption of two price. I have introduced the concept of a coherent risk measure and that each coherent risk measure can be represented by the distortion of a (normally distributed) probability variable at each acceptance level. I then showed how the distribution of option's payoffs is distorted by the Wang transformation in the Black-Scholes model, but in subsequent calculations I relied on the deduction of Karimov 2017.

Then I estimated the illiquidity or distortion parameter on real data. I also performed the calculations with four different tau parameters. I "compressed" the gamma parameter of options with different strikes but the same type and maturity into a single illiquidity parameter using two methods (ATM and weighted gamma). The relationship between the latter two parameters and absolute returns was examined using a VAR model.

My hypothesis was that there is a significant negative relationship between the lagged values of the gamma parameter and the current period returns. The VAR models indicate that there is no significant relationship between the lagged values of the gamma parameter and the absolute returns in the current period. I reject this hypothesis based on the VAR models.

My research question was whether it is possible to predict extreme declines based on the estimated gamma parameter. This question was only partially answered, as I did not investigate the relationship between the extreme gamma parameter and extreme declines, but the relationship between the gamma parameter and returns. On the other hand, the period under study did not include any extreme downturns, so the method was only tested under normal market conditions. In order to answer the research question, in addition to overcoming the limitations, it is necessary to improve the model (mainly the representation of the time series of the exchange rate).

4.1 Limitations

Despite the fact that I have collected the data from the Bloomberg, I find the model's main limitation to be the input data. On the one hand, the trading volume of the options under study was very low, we could see that in most of the observations there was no last price available. Also, the data series contained a number of outliers which can significantly distort the gamma parameter values, since in the case of a large tau, one outlier distorts the rest of the time series as well. On the other hand, I used daily data for both the underlying prices and the option prices, so there is no guarantee (in case of low option trading volume) that the underlying price was the last price when the option was traded, so the implied volatility could be distorted and the degree of this inaccuracy is increased by the fact that I calculated implied volatility on days when the option did not have a last price (I replaced it with a mid price).

4.1.1 Possibilities for improvement

The primary development option is to use higher frequency data in the estimation of the illiquidity parameter. It is recommended to remove outliers as they have a large impact on the calibrated gamma (also in the long run). On the other hand, we have seen that the Black-Scholes model itself has a number of limitations, the most influential in our case is the assumption of GBM dynamics regarding the underlying product price process. It may be useful to test other option pricing models (such as the Kou model) and other distortion functions.

5 Appendix A - Python codes of financial calculations

5.1 financial calculations

All the data used in the calculations are available at https://github.com/pfurjesz/Szakdolgozat_2022. **Appendix A - Python codes for financial calculations 7.2.1 - 7.2.10** input data as **option_v1_csv**. And the input data for the **Appendix A - Python codes for financial calculations / 7.2.11. Generating at-the-money and weighted gamma time series** is named **final_v3**.

5.1.1 Used libraries

```
import warnings
warnings.filterwarnings('ignore')
from math import log, sqrt, pi, exp
from scipy.stats import norm
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import math as math
import os
import datetime
from datetime import datetime
from scipy.optimize import minimize_scalar
import scipy
N = norm.cdf  x norm lis eloszl s jel l se
```

5.1.2 Black-Scholes model

```
def BS_CALL(S_t, K, T, r, sigma):
    d1 = (np.log(S_t/K) + (r + sigma**2/2)*T) / (sigma*np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    return S_t * N(d1) - K * np.exp(-r*T)* N(d2)

def BS_PUT(S_t, K, T, r, sigma):
    d1 = (np.log(S_t/K) + (r + sigma**2/2)*T) / (sigma*np.sqrt(T))
    d2 = d1 - sigma* np.sqrt(T)
    return K*np.exp(-r*T)*N(-d2) - S_t*N(-d1)
```

5.1.3 Conic Black-Scholes model

```
def Kamirov_Black_Sholes_call(S_t,sigma,K,r,gamma,T):
    #bid
    d1_bid_call=(log(S_t/K)+(r+(1/2)*sigma**2.)*T-gamma*sigma*math.sqrt(T))/(sigma*math.sqrt(T))
    d2_bid_call=d1_bid_call-sigma*math.sqrt(T)
    bid_price=S_t*math.exp(-gamma*sigma*math.sqrt(T)) * N(d1_bid_call) - math.exp(-r * T)* K * N(d2_bid_call)
    #ask
    d1_ask_call=(log(S_t/K)-(r+1/2*sigma**2.)*T+gamma*sigma*math.sqrt(T))/(sigma*math.sqrt(T))
    d2_ask_call=d1_ask_call-sigma*math.sqrt(T)
    ask_price=S_t*math.exp(gamma*sigma*math.sqrt(T)) * N(d1_ask_call) - math.exp(-r * T)* K * N(d2_ask_call)
    return(bid_price,ask_price)

def Kamirov_Black_Sholes_put(S_t,sigma,K,r,gamma,T):
    d1_bid_put=(log(K/S_t)-(r+1/2*sigma**2.)*T-gamma*sigma*math.sqrt(T))/(sigma*math.sqrt(T))
    d2_bid_put=d1_bid_put+sigma*math.sqrt(T)
    bid_price=math.exp(-r*T)*K*N(d2_bid_put)-S_t*math.exp(gamma*sigma*math.sqrt(T))*N(d1_bid_put)
    d1_ask_put=(log(K/S_t)-(r+1/2*sigma**2.)*T+gamma*sigma*math.sqrt(T))/(sigma*math.sqrt(T))
    d2_ask_put=d1_ask_put+sigma*math.sqrt(T)
    ask_price=math.exp(-r*T)*K*N(d2_ask_put)-S_t*mah.exp(-gamma*sigma*math.sqrt(T))*N(d1_ask_put)
    return(bid_price,ask_price)
```

5.1.4 Approximation of implied volatility

```
def implied_vol(real, S_t, K, T, r, type):

    def call_obj(sigma):
        return (BS_CALL(S_t, K, T, r, sigma) - real)**2

    def put_obj(sigma):
        return (BS_PUT(S_t, K, T, r, sigma) - real)**2

    if type == 'call':
        res = minimize_scalar(call_obj, bounds=(0.01,6), method='bounded')
        return res.x
    elif type == 'put':
        res = minimize_scalar(put_obj, bounds=(0.01,6),
                              method='bounded')
        return res.x
    else:
        raise ValueError("type_ must be 'put' or 'call'")
```

5.1.5 Calibration of gamma parameter

```
def gamma_calibrator(real_bid_vec, real_ask_vec, K, T_vec,real_S_t_vec,IV_vec, r, type):

    def call_obj(gamma):
        TSE=0
        for k in range (0,len(real_bid_vec)):
            T=T_vec[k]
            S_t=real_S_t_vec[k]
            sigma=IV_vec[k]
            real_ask=real_ask_vec[k]
            real_bid=real_bid_vec[k]
            estimated_bid=Kamirov_Black_Sholes_call(S_t,sigma,K,r,gamma,T)[0]
            estimated_ask=Kamirov_Black_Sholes_call(S_t,sigma,K,r,gamma,T)[1]
            TSE=TSE+(((real_bid-estimated_bid)**2)+((real_ask-estimated_ask)**2))

        return TSE

    def put_obj(gamma):
        TSE=0
        for k in range (0,len(real_bid_vec)):
            T=T_vec[k]
            S_t=real_S_t_vec[k]
            sigma=IV_vec[k]
            real_ask=real_ask_vec[k]
            real_bid=real_bid_vec[k]
            estimated_bid=Kamirov_Black_Sholes_put(S_t,sigma,K,r,gamma,T)[0]
            estimated_ask=Kamirov_Black_Sholes_put(S_t,sigma,K,r,gamma,T)[1]
            TSE=TSE+(((real_bid-estimated_bid)**2)+((real_ask-estimated_ask)**2))
        return TSE

    if type == 'call':
        res = minimize_scalar(call_obj, bounds=(0.001,100), method='bounded')
        return res.x
    elif type == 'put':
        res = minimize_scalar(put_obj, bounds=(0.001,100),
                              method='bounded')
        return res.x
    else:
        raise ValueError("A tipusnak callnak vagy putnak kell lennie")
```

5.1.6 Data cleaning

```
def cleaner(Option_SPX):
    selector=[]
    Option_SPX["average"]=""
    for i in range(0,Option_SPX.shape[0]):
        if Option_SPX["price_type"].iloc[i]=="last" and np.isnan(Option_SPX["price"].iloc[i].item())==True:

            loc_bid=np.where((Option_SPX['date']==Option_SPX["date"].iloc[i]) &
                                                       (Option_SPX['type']==Option_SPX["type"].iloc[i]) &
                                                       (Option_SPX['strike']==Option_SPX["strike"].iloc[i]) &
                                                       (Option_SPX['price_type']=="bid"))[0][0]

            same_bid=Option_SPX["price"].iloc[loc_bid].item()

            loc_ask=np.where((Option_SPX['date']==Option_SPX["date"].iloc[i]) &
                                                       (Option_SPX['type']==Option_SPX["type"].iloc[i]) &
                                                       (Option_SPX['strike']==Option_SPX["strike"].iloc[i]) &
                                                       (Option_SPX['price_type']=="ask"))[0][0]

            same_ask=Option_SPX["price"].iloc[loc_ask].item()

            if np.isnan(same_bid) or np.isnan(same_ask):
                pass
            else:
                Option_SPX["price"][i]=(same_ask+same_bid)/2
                Option_SPX["average"][i]=1
                selector=selector+[loc_ask,loc_bid,i]

        elif Option_SPX["price_type"].iloc[i]=="last" and np.isnan(Option_SPX["price"].iloc[i].item())==False:


            loc_bid=np.where((Option_SPX['date']==Option_SPX["date"].iloc[i]) &
                                                       (Option_SPX['type']==Option_SPX["type"].iloc[i]) &
                                                       (Option_SPX['strike']==Option_SPX["strike"].iloc[i]) &
                                                       (Option_SPX['price_type']=="bid"))[0][0]

            same_bid=Option_SPX["price"].iloc[loc_bid].item()

            loc_ask=np.where((Option_SPX['date']==Option_SPX["date"].iloc[i]) &
                                                       (Option_SPX['type']==Option_SPX["type"].iloc[i]) &
                                                       (Option_SPX['strike']==Option_SPX["strike"].iloc[i]) &
                                                       (Option_SPX['price_type']=="ask"))[0][0]

            same_ask=Option_SPX["price"].iloc[loc_ask].item()

            if np.isnan(same_bid) or np.isnan(same_ask):
                pass
            else:
                Option_SPX["average"][i]=0
                selector=selector+[loc_ask,loc_bid,i]
    Option_SPX=Option_SPX.iloc[selector].reset_index(drop=True)
    return(Option_SPX)
```

5.1.7 Moneyness calculation

```
def moneyness(option_df,s_df):
    option_df["moneyness"]=""
    for i in range(0,option_df.shape[0]):
        date=option_df["date"].iloc[i]
        strike=option_df["strike"].iloc[i].item()
        spot=s_df["last"].loc[s_df["date"] == date].item()
        option_df["moneyness"][i]=spot/strike
    return(option_df)
```

5.1.8 Approximation of gamma parameter on real data

```
def gamma_estimator(K,ctype,tau,r,spx,spx_option): # row zero min date
    tech_param=0
    out = pd.DataFrame(columns=['date','SP','real_last','real_bid', 'real_ask', 'BS_price',
                                'gamma','k_bid','k_ask','average'])

    K_selected=spx_option.loc[spx_option['strike'] == K]
    type=K_selected.loc[spx_option["type"] == ctype]
    last=type.loc[type['price_type'] == "last"]
    last=last.reset_index(drop=True)
    bid=type.loc[type['price_type'] == "bid"]
    bid=bid.reset_index(drop=True)
    ask=type.loc[type['price_type'] == "ask"]
    ask=ask.reset_index(drop=True)
    bid_tau=[None] * tau
    ask_tau=[None] * tau
    IV_tau=[None] * tau
    T_tau=[None] * tau
    S_t_tau=[None] * tau
    last_tau=[None] * tau
    average_tau=[None] * tau
    for i in range(tau,last.shape[0]):
        date=last["date"].iloc[i]
        for j in range (0,tau):
            bid_tau[j]=bid["price"].iloc[i-j].item()
            ask_tau[j]=ask["price"].iloc[i-j].item()
            IV_tau[j]=last["IV"].iloc[i-j]
            T_tau[j]=last["T"].iloc[i-j]
            loc_SPX=np.where((SPX['date']==date))[0][0]
            S_t_tau[j]=SPX["last"].iloc[loc_SPX-j]
            last_tau[j]=last["price"].iloc[i-j].item()
            average_tau[j]=last["average"].iloc[i-j]
        #breakpoint()
        gamma=gamma_calibrator(bid_tau, ask_tau, K, T_tau,S_t_tau,IV_tau, r, ctype)
        if ctype=="call":
            BS_price=BS_CALL(S_t_tau[0], K, T_tau[0], r, IV_tau[0])
            bid_est=Kamirov_Black_Sholes_call(S_t_tau[0],IV_tau[0],K,r,gamma,T_tau[0])[0]
            ask_est=Kamirov_Black_Sholes_call(S_t_tau[0],IV_tau[0],K,r,gamma,T_tau[0])[1]
        elif ctype=="put":
            BS_price=BS_PUT(S_t_tau[0], K, T_tau[0], r, IV_tau[0])
            bid_est=Kamirov_Black_Sholes_put(S_t_tau[0],IV_tau[0],K,r,gamma,T_tau[0])[0]
            ask_est=Kamirov_Black_Sholes_put(S_t_tau[0],IV_tau[0],K,r,gamma,T_tau[0])[1]
        out.loc[len(out)] = [date,S_t_tau[0],last_tau[0],bid_tau[0],ask_tau[0],BS_price,gamma,bid_est,ask_est,aver
    return(out)
```

5.1.9 Approximation of implied volatility on real data

```
def BS_IV(Option_SPX,r):
    Option_SPX["IV"]=""
    Option_SPX["T"]=""
    for i in range(0,Option_SPX.shape[0]):
        K=Option_SPX["strike"].iloc[i].item()
        date=Option_SPX["date"].iloc[i]
        S_t=SPX["last"].loc[SPX['date'] == date].item()
        type=Option_SPX["type"].iloc[i]
        time_to_maturity=time_of_maturity-date
        Option_SPX["T"].iloc[i]=time_to_maturity.days/360
        T=time_to_maturity.days/360
        real=Option_SPX["price"].iloc[i].item()
        Option_SPX["IV"].iloc[i]=implied_vol(real, S_t, K, T, r, type)
    return(Option_SPX)
```

5.1.10 Calculation of simple gamma time series

```
out_main=[]
gamma_type="backward"
strikes=[3700,3725,3750,3775,3800,3825,3850,3875,3900,3925,3950,3975,4000,4025,4050,4075,4100,4125,4150,4175]
types=["call","put"]
r=0.02
taus=[1,7,14,28]
tech_param=0
out = pd.DataFrame(columns=['date','SP','real_last','real_bid', 'real_ask', 'BS_price','gamma','k_bid','k_ask','a
for type in types:
    for strike in strikes:
        K=strike
        for tau in taus:
            out=gamma_estimator(K,type,tau,r,SPX,Option_SPX)
            out["K"]=""
            out["type"]=""
            out["tau"]=""
            out["K"]=K
            out["type"]=type
            out["tau"]=tau
            if tech_param==0:
                out_main=out
                tech_param=1
            else:
                out_main=out_main.append(out)
out_main.to_excel("final_v3.xlsx",sheet_name='Sheet_name_1')
```

5.1.11 Calculation of at-the-money gamma and weighted gamma time series

Due to the technical nature and length of the calculations, the codes are available at https://github.com/pfurjesz/Quantitative-Finance-Admission.git as **ATM gamma and weighted gamma calculation**. Input data for statistical calculations are also available as **all_wput , allw_call, all_ATMcall** and **all_ATMput**.

6 Appendix B - Results of statistical tests and calculations

6.0.1 ATM call gamma - VAR model results

	Call tau 1	**call tau 7**	**Call tau 14**	**Call tau 2**
ADF-test (p-value)	0.01	0.02	0.21	0.49
ADF-test diff (p-value)	NA	0.01	0.01	0.01
AIC(n)	2.00	1.00	1.00	1.00
HQ(n)	1.00	1.00	2.00	1.00
SC(n)	1.00	1.00	1.00	1.00
FPE(n)	1.00	1.00	1.00	1.00
s&p500 - gamma adjusted R square	-0.02	-0.01	-0.01	-0.01
s&p500 - gamma F test p value	0.85	0.60	0.83	0.83
gamma - s&p500 adjusted R square	0.06	0.03	0.12	0.07
gamma - s&p500 F-test p value	0.01	0.07	0.00	0.01

6.0.2 ATM put gamma - VAR VAR model results

	Put tau 1	**Put tau 7**	**Put tau 14**	**Put ta**
ADF-test (p-value)	0.01	0.17	0.25	0.73
ADF-test diff (p-value)	NA	0.01	0.01	0.01
AIC(n)	1.00	1.00	2.00	1.00
HQ(n)	1.00	2.00	2.00	1.00
SC(n)	1.00	1.00	1.00	1.00
FPE(n)	1.00	1.00	2.00	1.00
s&p500 - gamma adjusted R square	-0.01	-0.01	0.01	-0.01
s&p500 - gamma gamma F test p value	0.73	0.82	0.24	0.62
gamma - s&p500 adjusted R square	0.00	0.12	0.01	0.21
gamma - s&p500 gamma F test p value	0.60	0.00	0.00	0.00

	Call tau 1	call tau 7	Call tau 14	C
ADF-test (p value)	0.02	0.25	0.43	0
ADF-test diff (p value)	NA	0.01	0.01	0
AIC(n)	1.00	1.00	1.00	1
HQ(n)	1.00	2.00	2.00	2
SC(n)	1.00	1.00	1.00	1
FPE(n)	1.00	1.00	1.00	2
s&p500 - gamma gamma adjusted R square	-0.01	0.00	-0.01	-0
s&p500 - gamma F test p value	0.83	0.32	0.83	0
gamma - s&p500 adjusted R square	0.00	0.12	0.01	0
gamma - s&p500 F test p value	0.33	0.00	0.22	0

6.0.3 Weighted call gamma - VAR model results

6.0.4 Weighted put gamma - VAR model results

	Put tau 1	Put tau 7	Put tau 14	Put tau 2
ADF-test (p value)	0.02	0.25	0.43	0.32
ADF-test diff (p value)	NA	0.01	0.01	0.01
AIC(n)	1.00	1.00	1.00	1.00
HQ(n)	1.00	2.00	1.00	2.00
SC(n)	1.00	1.00	1.00	1.00
FPE(n)	2.00	2.00	1.00	1.00
s&p500 - gamma adjusted R square	-0.01	0.00	-0.01	-0.01
s&p500 - gamma F test p value	0.83	0.32	0.83	0.83
gamma - s&p500 adjusted R square	0.00	0.12	0.01	0.01
gamma - s&p500 F test p value	0.33	0.00	0.22	0.22

7 References

Basics of conic finance

F. Black, M. Scholes (1973). “The Pricing of Options and Corporate Liabilities”. In: *Journal of Political Economy* 81, pp. 637–654. URL: https://www.jstor.org/stable/1831029#metadata_info_tab_contents.

P. Artzner, F. Delbaen J.M. (1999). “COHERENT MEASURES OF RISK”. In: *Mathematical Finance* 9, pp. 03–228. URL: https://people.math.ethz.ch/~delbaen/ftp/preprints/CoherentMF.pdf.

Kusuoka, S. (2001). “On law invariant coherent risk measures”. In: *Advances in Mathematical Economics* 3, pp. 83–95. URL: https://link.springer.com/chapter/10.1007/978-4-431-67891-5_4.

Wang, Z. (2002). “Discussion”. In: *The Journal of Finance* 4, pp. 1240–1245. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/0022-1082.00366.

Main publications of conic finance

A. S. Cherny, D. B. Madan (2006). “Pricing and Hedging in Incomplete Markets with Coherent Risk”. In: *SSRN*. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=904806.

— (2008). “New Measures for Performance Evaluation”. In: *SSRN*. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=955472.

— (2010). “MARKETS AS A COUNTERPARTY: AN INTRODUCTION TO CONIC FINANCE”. In: *International Journal of Theoretical and Applied Finance* 13, pp. 1149–1177. URL: https://www.worldscientific.com/doi/10.1142/S0219024910006157.

D. Madan W, SCHOUTENS (2016). *APPLIED CONIC FINANCE*. Contributions to Management Science. Springer. ISBN: 978-1-107-15169-7.

Applications of conic finance

Karimov, A. (2017). *Identifying Stock Market Bubbles*. Contributions to Management Science. Springer. ISBN: 978-3-319-65008-1.

Z. Lia, W. Zhangb (2019). "An analytical approximation approach for pricing European options in a two-price economy". In: URL: https://www.sciencedirect.com/science/article/abs/pii/S1062940818306065.

M. Michielon, A. Khedher and P. Spreij (2021). "Liquidity-free implied volatilities: an approach using conic finance". In: URL: https://arxiv.org/abs/2110.11718.

W. Wang, Xiaoping H. (2022). "Pricing American Options by a Fourier Transform Multinomial Tree in a Conic Market". In: URL: https://doi.org/10.1155/2022/8650500.