The Black-Scholes-Merton Model

PC Lab in Finance

Exercise 1

Prof. Dr. Ing. Rainer Schöbel and Lina Kalimulina

Summer Term 2016

Konstantin Smirnov, Student-ID: 3980253, Economics and Finance

Tübingen, April 20, 2016

1 B-S-M model for European options

The goal of this exercise is the determination of the fair price of a put option with given parameters of dataset 2 by using the BSM-Model and, furthermore, the general derivation of the options price as well as a selection of Greeks as a graphical representation for varying T and S.
The general run-file for this exercise is called *"Task1_Master"*. Every corresponding subtask of this exercise can be called in this the corresponding subsection.

1.A Calculations

To calculate the fair price, the lower and upper bound and the Delta, the Gamma, the Vega and the Theta of the put option following formulas were used:
The Black-Scholes-Merton-Formula for a call option:

$$c(S_t, t) = S_t e^{(h-r)(T-t)} N(d_1) - X e^{(h-r)(T-t)} N(d_2) \tag{1}$$

and respectively for a put option:

$$p(S_t, t) = X e^{(h-r)(T-t)} N(-d_2) - S_t e^{(h-r)(T-t)} N(-d_1) \tag{2}$$

with

$$d_1 = \frac{ln(\frac{S_t}{X}) + (h + 0.5\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \tag{3}$$

and

$$d_2 = d_1 - \sigma\sqrt{T-t} \tag{4}$$

We follow the notational convention of:

- c = fair price of a European call
- p = fair price of a European put
- S = price of the underlying in t
- X = exercise price
- σ = volatility of the underlying
- h = r - q in the continous dividend case

The Lower Bounder is defined as:

$$\tag{5}$$

For the Greeks I used following equations:

Delta

$$\Delta_p(S_t, t) = e^{(h-r)(T-t)}(N(d_1) - 1) < 0 \tag{6}$$

Gamma

$$\Gamma(S_t, t) = \frac{e^{(h-r)(T-t)}N'(d_1)}{S_t\sigma\sqrt{T-t}} > 0 \tag{7}$$

Vega

$$\mathcal{V}(S_t, t) = S_t\sqrt{T-t}N'(d_1)e^{(h-r)(T-t)} > 0 \tag{8}$$

Theta

$$\Theta_p(S_t, t) = -\frac{S_t\sigma N'(d_1)e^{(h-r)(T-t)}}{2\sqrt{T-t}}(h-r)S_t e^{(h-r)(T-t)}N(-d_1) - Xe^{(h-r)(T-t)}N(-d_2) \tag{9}$$

The results are:

Price	**Lower Bound**	**Upper Bound**	Δ	Γ	Θ	$\mathcal{V}$
2.6014	0	64.5143	-0.3024	0.0284	-8.4357	12.1626

Table 1: Calculated Values for Dataset # 2

Technical notes on task 1A:
A function for each corresponding calculation was created named respectively *Bounds.m, BSM_Put_Task1, Greeks.m*. Each function contains the corresponding formula for each calculation. To keep the code universal the the probability the *norm_dist.m* file was programmed.

1.B Calculations

To analyze the dynamics of the put option with respect to a changing stock price and time to maturity, 2D graphs were set up. The graphs can be called by using the functions *BSM_Graph_Stock.m* and *BSM_Graph_Stock.m* in the task 1B part of the main script. Note that to plot the graphs calculations of subtask 1A has to be called before plotting.
Figure 1 shows the dynamic of a stock price change on value and the upper & lower bound of the put option holding the other parameters constant. The results are in line with the theory in a BSM-Put framework: As you can clearly see in the yellow line the value of the put is decreasing while the stock price is increasing. The blue line shows the lower bound of the option. It corresponds to the price of the put, but has a more abrupt change due to equation (2). The upper bound represented by the red line does not change since its not affected by the stock price.

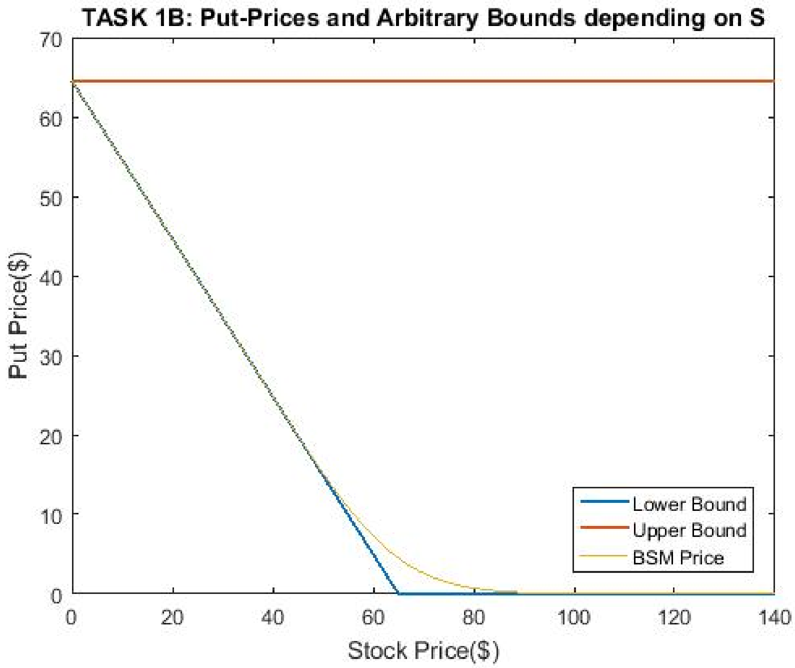

Figure 1: BSM-Price, Lower & Upper Bound depending on stock price

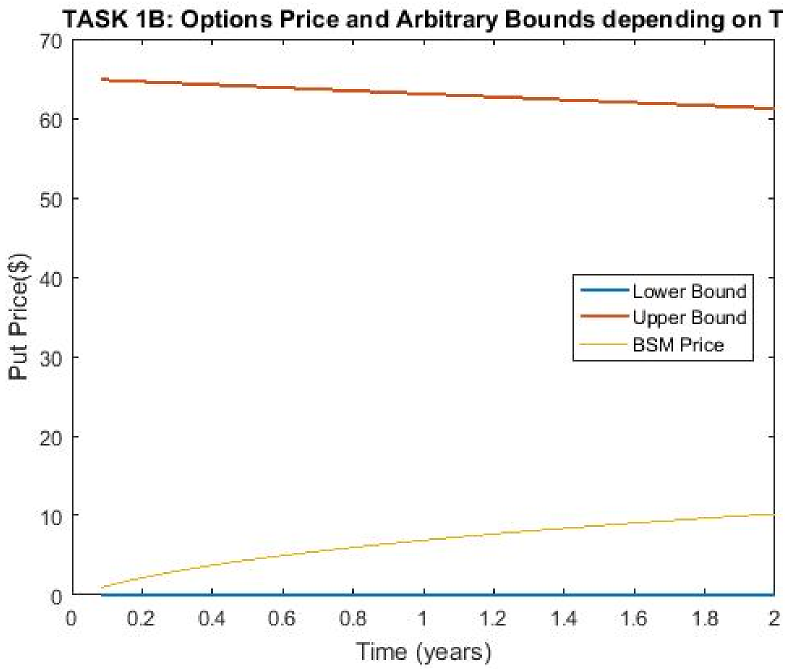

Figure 2: BSM-Price, Lower & Upper Bound depending on time to maturity

Figure 2 illustrates the dynamics of the put option with an increasing time to maturity. The upper bound is slowly downward slopping since the cost of carry increases for a longer time to maturity. The value of the put is increasing over time due to the derivative's "downward protection": For a given volatility a holder of a put can benefit from downward movements, while not suffering losses through downturns by not exercising the option. A shorter time to maturity implies that there is more time for potential gains. Due to dataset #2 the lower bound stays constant with a value of zero since we are in the out of money framework.

Technical Notes on 1B:
2D Graphs were plotted creating a series for S 1$ to $140 and respectively a time series for time to maturity for 2 years. To plot the upper and lower bounds it was necessary to create and vector as a plot. For this purpose the input variable *"S_Lenght"* was created which contains the vectors *"S_Grid"* and *"T_Grid"* representing the X-Axis as well as the input arguments for each single calculation.

1.C 3D-Graphs

To analyze the dynamics of the put in more detail, the focus in this subtask was to create 3D-plots for the calculated put price and for the Greeks Delta, Gamma, Theta and Vega depending changing time to maturities and stock prices. Each figure can be plotted by calling the corresponding function in the subsection of the mainscript *"Task1_Master"*. Note again that for plotting calculations of subtask 1A have to be called.

Figure 3 proves theory represented in Task 1B. It can clearly be seen that the option price decreases when the price of the underlying increases and vice versa for time to maturity due to the mentioned reasons. However, we can observe an inverted structure when the put is extremely deep in the money at very short maturities (the red-colored area). At this point being knocked out from in-the-money is very unlikely.

Delta is defined as the sensitivity of the option value with respect to changes in the respective underlying represented in Figure 4 The value of the put option increases simultaneously with a decrease in the stock price and therefore the delta is always negative. Moreover, for very short maturities we observe that the Delta tends reach the smallest value, therefore the deeper the put is in the money prices combined with a very short time to maturity, the value of the put is most sensitive to changes in the stock prices represented in the deep blue area of the plot.

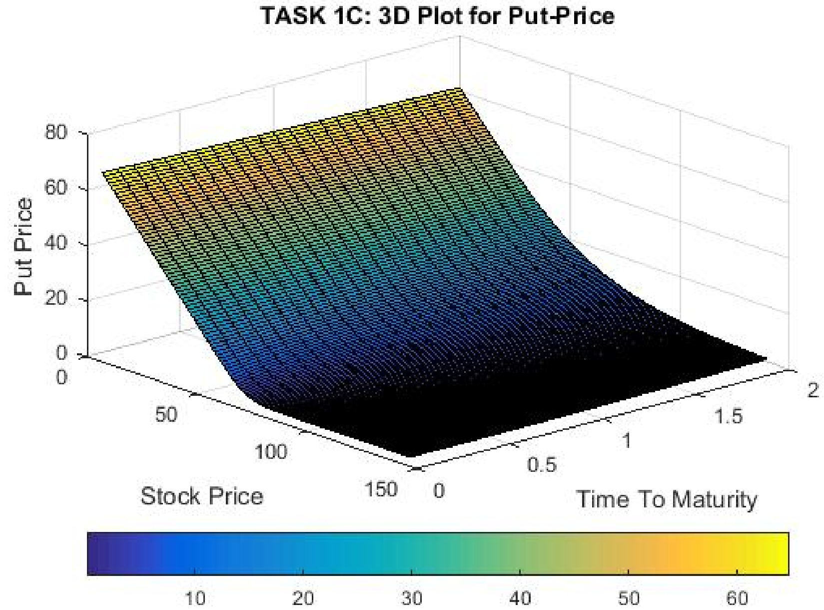

Figure 3: Value of the put depend on stock price & time to maturity

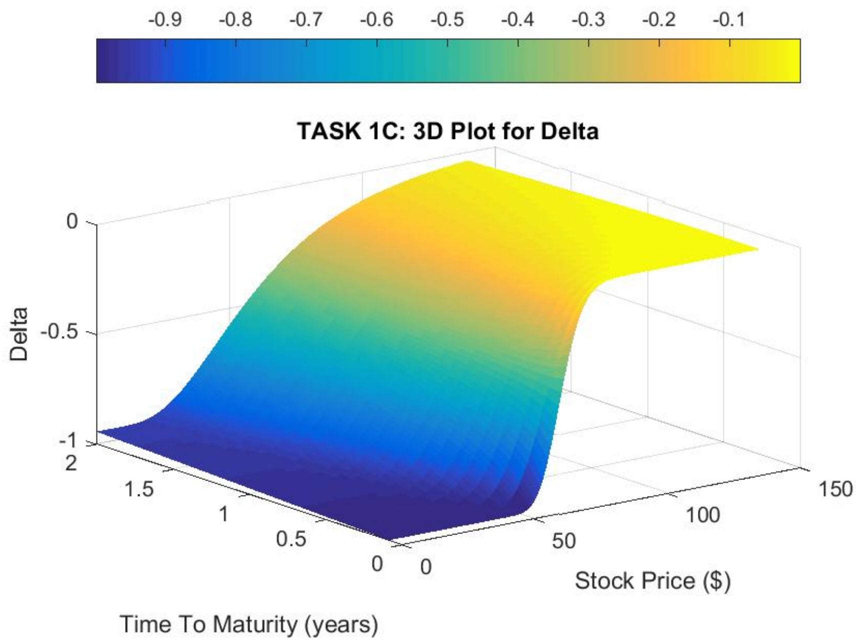

Figure 4: Value of Delta depend on stock price & time to maturity

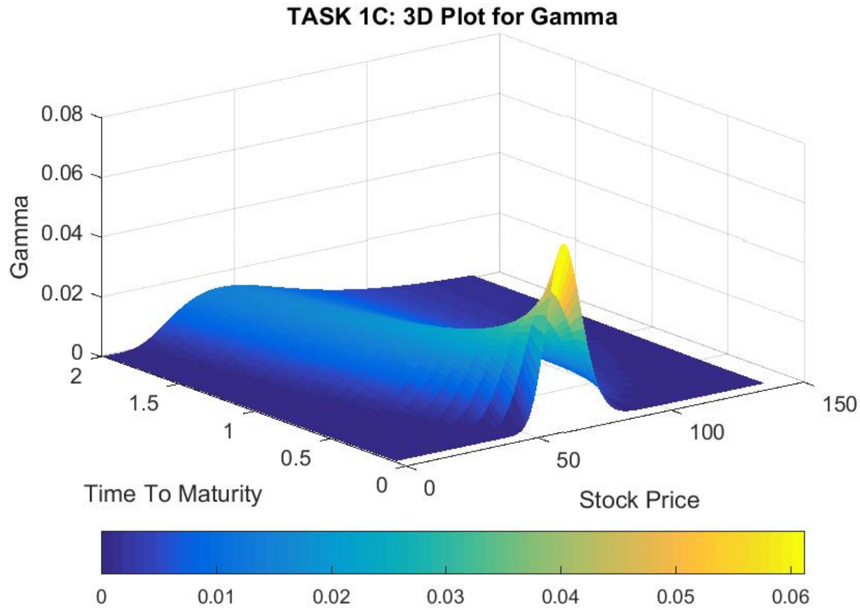

Figure 5: Value of the Gamma depend on stock price & time to maturity

The Gamma of a put option shows how quickly the delta changes if the stock price changes (which itself is the first partial derivative of delta), represented in Figure 5. The sensitivity of the option value to changes in the stock price tends to be more severe in the region around at the money, since value of the option is the most likely to change. Moreover, since Delta converges faster for short maturities, the gamma tends to have the steepest distribution around the strike for very short maturities represented in the yellow area of this figure.

The Theta of a put option measures the change in the option's price as time to expiration decreases. As illustrated in Figure 6 you can observe again the steepest distribution of negative values in the deep blue area. In this area the put is at-the-money and has a very short time to maturity. This corresponds to the theory: The likeliness that the value of the put changes in that case is the most unlikely, therefore it has an negative impact of the value of the option.

The option's Vega measures the sensitivity of the option with respect to volatility and it is always positive. Also for Vega, illustrated in Figure 7 the highest sensitivity of the put price with respect to volatility can be observed at the money. In this case the put is again either about to become worthless or valuable, so volatility changes matter the most. On the other hand, the more time to maturity increases, the stronger this effect is. The probability that effects on the value happen at the very short maturity end is the most unlikely, thus the effect is smaller. Moreover, The vega always converges to 0 when the stock price is far from the strike price as the volatility impact is low.

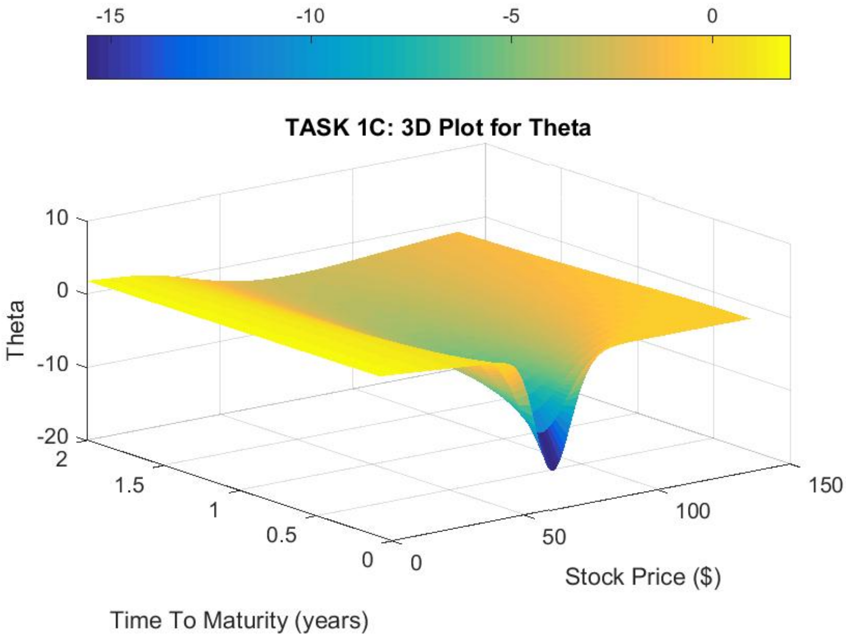

Figure 6: Value of the Vega depend on stock price & time to maturity

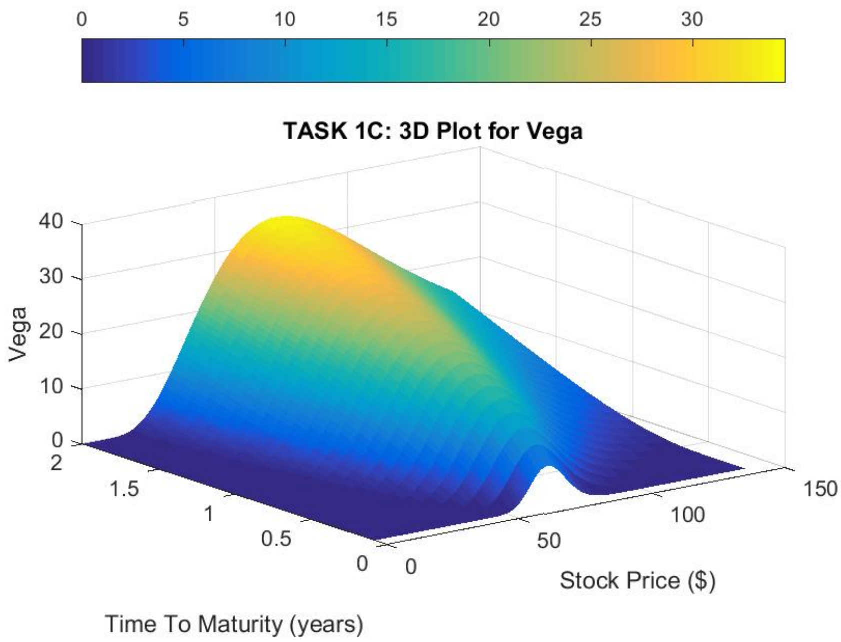

Figure 7: Value of the Vega depend on stock price & time to maturity

Technical notes on Task 1C:
I created special functions for each 3D plot due to keep a systematic overview for each Figure. Due to efficiency reasons I designed two matrices with help of the meshgrid command in order to create 2D-space with respect to S= 1 to 140 and Time to maturity up to 2 years To calculate the values of the put price and the "Greeks" the matrices were used as an input to the corresponding functions.

2 Calculation of implied interest rate and volatilities

The implied volatility is an import concept due to its practical application in Finance. While the standard theoretical BSM model assumes a flat volatility for different strikes, it will be shown that in practice this is not the case. Therefore the focus in this task lays on the calculation and illustration of the implied volatilities for an European ODAX Put with a time to maturity of June 2016 (data set #2). By calling the main-script *TASK2_Master.m* all necessary calculations are exercised. For this purpose it necessary to format the given PDF-data set into an excel sheet called *"StrikesandPricesJune2016.xlsm"* which implements the values for the observable strikes and settlement prices respectively for a call and a put. By applying the command *"xlsread"* the data set is imported into Matlab and saved as an matrix called *data*. For practical reasons I defined each column of this matrix separately (*X, put and call*).
To calculate the implied volatility I applied the following formula

$$V^{BSM}(S, X, r, q, T, \sigma_i mp(X, T)) = V^{Market}(S, X, r, q, T) \tag{10}$$

where V^{BSM} is the BSM option price and V^{Market} the observed price in the market.

2.A Calculation of the Implied Interest Rate

In order to calculate the implied volatility, I first calculated the implied interest rate out of the given data set. To do so I rearranged the Put-Call- Parity to get the following formula:

$$r_{imp} = -(\frac{ln\frac{P_t+S_t-C_t}{X}}{T-t})$$

For each Put and Call value with respect to the strike price you will get an individual implied interest rate. In our data set #2 137 implied interest rates were calculated. I decided to take the arithmetic average of this vector due to practical reasons. Besides this its the most obvious approach since there is no theoretical reason to take another method regarding the focus of this task. I get a slightly negative average implied interest rate of $r_{imp.av}$=**-0.0009053**. This negative value is reasonable due to the 0% interest rate condition we are facing today.

Technical notes on Task 2A:
By calling the function *r_implied_mean* the average implied interest will be calculated, whereas by calling the function *r_vec* it generates the general implied interest rates vector for the given values. The function itself just basically contains the mentioned formula.

2.B &

2.B.1 Calculation and Illustration of the Implied Volatility

Since Equation 10 for the implied volatility is mathematically not solvable I used the so-called numerical *"Newton-Raphson"* approach which generally calculates the root for every non-linear function.
The equations are:

$$\sigma_{imp}^{i+1} = \sigma_{imp}^{i} - \frac{f(\sigma_{imp})}{f'(\sigma_{imp})} \tag{11}$$

where

$$f(\sigma_{imp}) \equiv P^{BSM} - P^{Market} = 0 \tag{12}$$

with P^{BSM} theoretical BSM price and P^{Market} observed market price.

As long as equation 12 is above 0 this function iterates you the wanted root, in our case the implied volatility of data set #2. As you can see in Figure 8 the implied volatility of the European put is, in sharp contrast to the BSM model predictions, not flat and it moves randomly i.e. for any given strike the implied volatility can be either decreasing or increasing which draws in our case a structure which is called volatility skew. It usually appears for long-term equity options and index options. Since the moneyness of the put option is defined as X\S the implied volatility tends to be higher for strikes further away from the unity point, i.e. from at-the-money. With our input you will observe the classical *"Volatility Smile"* with its minimum at-the-money.

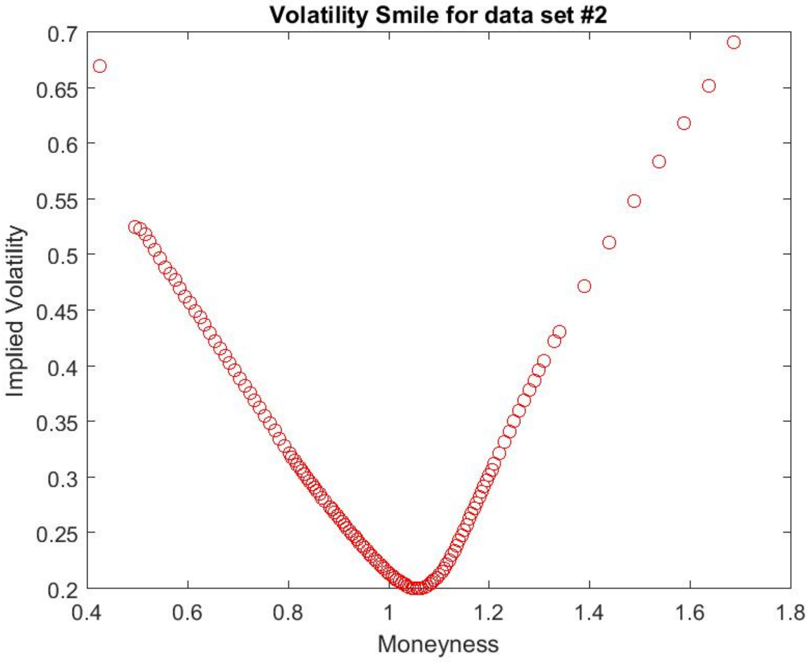

Figure 8: Volatility Smile

Technical notes on Task 2B:
In order to calculate the implied volatilty vector and the corresponding figure for the given data set I wrote a function called *implied_vola.m*. As you can see in equation 10 and 12, I had to implement two other functions *bsm_put_t2.m* and *vega.m* to calculate the theoretical BSM price and the vega. For the itretation process I set up an while loop with the end condition *tol* which has to be close to zero, as I already mentioned. As a starting value I defined σ=0.50 since higher volatilities are unrealistic. As long as the σ_{imp} is high enough the iteration will continue by taking the new calculated σ_{imp} for the next iteration. This vector is then simply plotted with respect to moneyness. As you can see in the calculated vector the last 4 values will result in *infinity* and *NaN* values since the approach calculates extremely low and close to zero implied volatilities for highly improbable strikes or market conditions in general. For practical reasons I found this rather implausible and chose not to display them.