

Fixing Volatility – Assignment 6 – Jake Mulready

Introduction

The purpose of this study is to leverage data obtained from Yahoo Finance. The goal is to construct and analyze the implied volatility surface of options associated with the S&P 500 index. This entails gathering options data encompassing various maturity dates, allowing for a comprehensive understanding of volatility dynamics across different time horizons.

Data

- Data Sources: Standard options quotes for the S&P 500 (^SPX) obtained from yahoo finance.
- Took all available maturity dates from today till 2029.
- Assets Collected: Options data including strike prices, expiration dates, bid/ask prices, and IVs.

Methodology

I first gathered all the necessary raw data from yahoo finance. From this data I was able to manipulate it to get the option data that I needed, including moneyness, time to maturity (TTM), etc. I noticed that as TTM was increasing, it got more and more unreliable. Due to this I decided to instead only graph up to one year till maturity. After I finished the first implied volatility surface (Graph 1), I noticed how not smooth the surface was and therefore to get a better model, I needed to smooth it out. I did this by using a spline model. This is then represented in graph 2. The curvature of the surface seemed to fit business intuition, except that when maturity was decreasing, moneyness increasing, implied volatility kept getting more negative. This seemed like an error in the data, so I decided to take it out.

From here I needed to first model the Heston model. I did this by using the following equations to model it,

```
d1 = (np.log(S / K) + (r - q + theta) * T) / (sigma * np.sqrt(T))
d2 = d1 - sigma * np.sqrt(T)

# Integrate the variance process
integration_factor = (1 - np.exp(-kappa * T)) / (kappa * T)
variance = v0 * np.exp(-kappa * T) + theta * integration_factor

# Calculate the Heston model option price
price = S * np.exp(-q * T) * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
```

and then the following to calibrate it,

```
errors = []
for i in range(len(data)):
    S = data.iloc[i]['lastPrice'] # Stock Price
    K = data.iloc[i]['strike'] # Strike Price
    T = data.iloc[i]['time_to_maturity'] / 365.0 # Maturity (Years)
    r = 0.05 # Assuming a risk-free rate of 5%
    q = 0.0 # No dividend yield
    iv = data.iloc[i]['impliedVolatility'] # Implied Volatility
    kappa, theta, sigma, rho, v0 = params
    d1 = (np.log(S / K) + (r - q + 0.5 * v0) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    model_value = heston_model(params, S, K, T, r, q)
    market_value = S * np.exp(-q * T) * norm.cdf(d1) - K * np.exp(-r * T) *
norm.cdf(d2)
    error = iv - model_value / market_value
    errors.append(error)
```

Using an optimization function like the following I was able to get the best Heston Model parameters.

```
result = minimize(lambda params: np.sum(heston_calibration(params,
options_data)**2), initial_params, bounds=bounds)
```

This took around 5-10 min to run every time, so I decided to devise a way to get better initial parameters so that it does not take as long, but also so that it does not just converge to a local minima or maxima. After obtaining the Heston parameters, I plugged them back into the Heston model function to get prices based on the optimized parameters. I then plugged those prices, as well as the other parameters into the B&S model and looped through. This gave me new implied volatilities that are shown in graph 3 and 4.

Empirical Results

Our empirical results indicate that the Heston model, after optimizing, provided the following parameters:

```
Heston Model Parameters:  
Kappa: 2.0  
Theta: 1.0  
Sigma: 0.3  
Rho: -0.5  
V0: 1.0
```

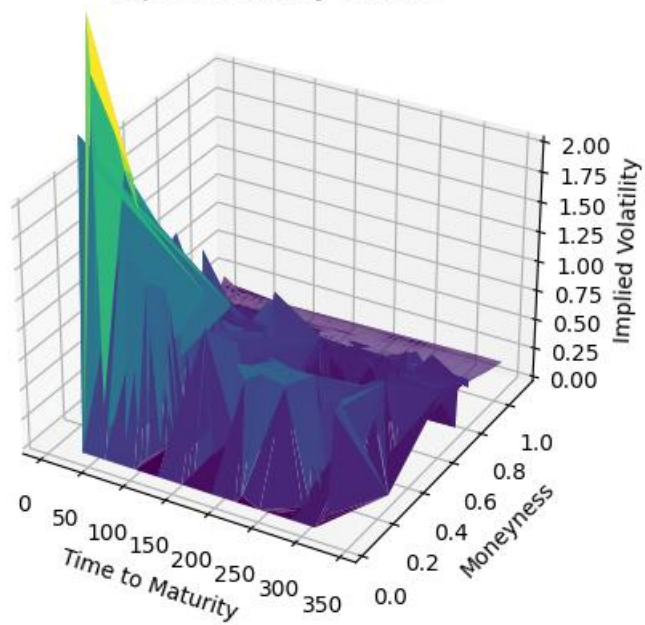
Entering these into the Heston model to obtain optimal option prices, we get a RMSE of 1038.9134509054938 so there is obviously an issue in the data or the way that I modeled it. I then plugged these prices into the B&S model giving graph 3, and a smoothed spline model (graph 4). The RMSE of the B&S volatilities was 60.95390083517594. This is also an unacceptable RMSE.

Conclusion

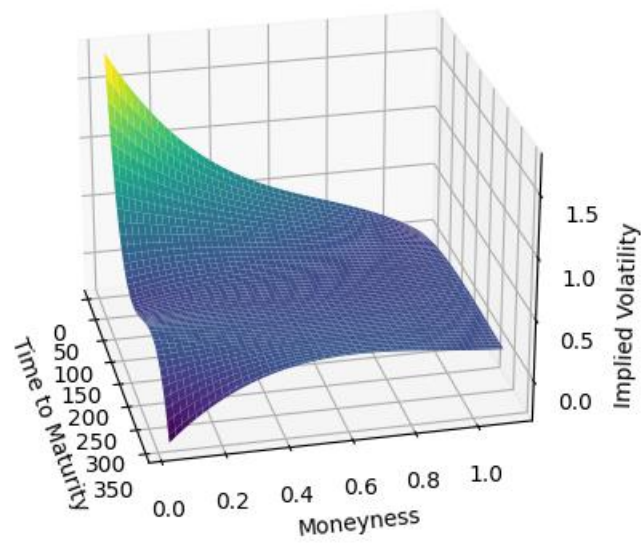
In conclusion, this study provides good insights into volatility dynamics and option pricing for the S&P 500. While the Heston model demonstrates potential for capturing complex volatilities, challenges in parameter estimation and computational efficiency remain. Further refinement of modeling practices

and data processing needs to be done to enhance model accuracy and reliability. Additionally, alternative modeling approaches and other checks could provide further improvement to our performance.

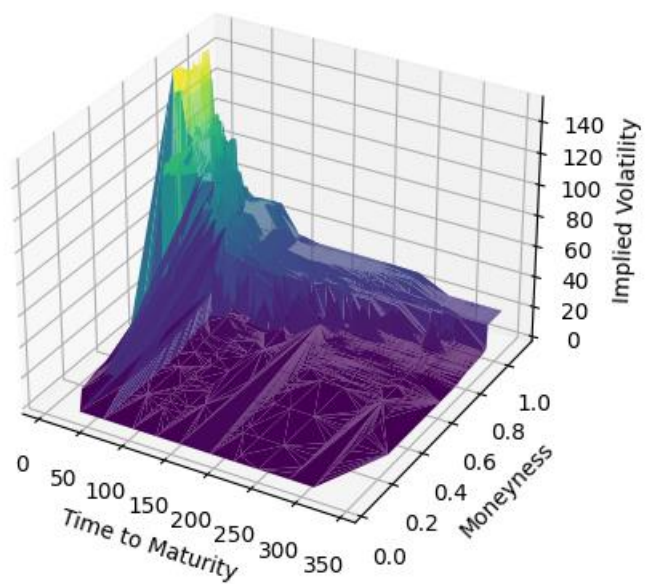
Implied Volatility Surface



Implied Volatility Surface



Original Black-Scholes Implied Volatility Surface



Smoothed Black-Scholes Implied Volatility Surface

