Approximating Implied Volatility of Call Options with the Black-Scholes Model using Newton-Raphson Method

Jason Au, Imanol (Manny) Benitez,
Zhuowei Deng, Hikaru Nagano, Eddy Rivera

March 18, 2024

Contents

1 Introduction

The implied volatility of an option contract is an important parameter, reflecting the market's expectation of future volatility based on a given security's option prices. It is a crucial component because it influences the premium paid or received for options. Higher implied volatility suggests a greater uncertainty about the security's future price, leading to higher option premiums, and vice versa. However, implied volatility cannot be directly observed or easily calculated. Therefore, numerical methods, such as root-finding methods, are useful for computing the implied volatility. With this in mind, our group decided to investigate the applicability of using the Newton-Raphson method, a commonly used root-finding technique, for determining the implied volatility of a given option.

2 Methodology

We chose to employ the Black-Scholes model of option pricing. This is a mathematical equation that estimates the theoretical value of derivatives based on five parameters: current stock price, strike price, time till expiration (in years), risk-free interest rate, and the market price of the call option. The goal is for this equality to be true: $Call_{BS}(\sigma_{implied}) = P$, which is when the Black-Scholes call option function as a function of implied volatility, with everything else held constant, equals the price of the call option contract, P. We do this by finding the root of the function $f(x) = Call_{BS}(\sigma_{implied}) - P$.

2.1 Mathematical formulas for the Black-Scholes Model

$$Call_{BS}(\sigma_{implied}) = N(d_1)S - N(d_2)Ke^{-rT} \tag{1}$$

$$d_1 = \frac{\ln \frac{S}{K} + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \tag{2}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{3}$$

$Call_{BS}(\sigma_{implied})$ = call option price
N = cumulative distribution function
T = maturity (in years)
S = stock price
K = strike price
r = risk free rate
σ = volatility

2.2 Mathematical formulas for the Newton-Raphson Method

Newton's method employs derivatives to iteratively refine the estimate until it converges to a root. Start with p_0 close to p, then do

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{1}$$

2.3 Implementation in MATLAB

```
S = 162.83; % Current stock price
K = 165; % Strike price
T = 0.0685; % Time to expiration (1 year)
r = 0.04; % Risk-free interest rate (4%)
C = 8.39; % Observed market price of the option
impliedVol = findImpliedVolatility(S, K, T, r, C);

function impliedVol = findImpliedVolatility(S, K, T, r, C)

   % Initial guess for implied volatility
   sigma = 0.3;

   % Convergence threshold
   threshold = 0.001;

   % Iteration counter
   iteration = -1;
   max_iterations = 10

   % Arrays to store each step's values
   steps = [];
   volatilities = [];
   d1_values = [];
   d2_values = [];
   prices = [];
   vegas = [];

   while  iteration < max_iterations
       iteration = iteration + 1; % Increment iteration counter

       [theoretical_price, d1, d2, dVega] = black_scholes_call(S, K, T, r, sigma);

       % Calculate the difference between theoretical and observed prices
       price_diff = theoretical_price - C;
       % Check for convergence
       if abs(price_diff) < threshold % Check for convergence
           break;
       end

       % Store values
       steps(end+1) = iteration;
       volatilities(end+1) = sigma;
       d1_values(end+1) = d1;
       d2_values(end+1) = d2;
       prices(end+1) = theoretical_price;
       vegas(end+1) = dVega;
```

```
        sigma = sigma - (price_diff / dVega); % Update sigma
    end
    impliedVol = sigma;

    % Display table
    T = table(steps', volatilities', d1_values', d2_values', prices', vegas', ...
        'VariableNames', {'Step', 'Volatility', 'd1', 'd2', 'Fair Value', 'Vega'});
    disp(T);
end

function [price, d1, d2, dVega] = black_scholes_call(S, K, T, r, sigma)
    d1 = (log(S / K) + (r + 0.5 * sigma^2) * T) / (sigma * sqrt(T));
    d2 = d1 - sigma * sqrt(T);
    price = -S * normcdf(-d1) + K * exp(-r * T) * normcdf(-d2);
    dVega = S * normpdf(d1) * sqrt(T); % Vega calculation
    % Return d1, d2, and Vega for storage
end
```

In the code, we utilize an initial guess for the implied volatility, and, based on that, calculate the theoretical price of the option. We then adjust the implied volatility iteratively by subtracting the value of $f(x)$ divided by the derivative of the function from the implied volatility. This is repeated until the difference of the implied volatility between two iterations is less than a certain error value (in this case we used 0.001). The code outputs the estimated implied volatility for each iteration, and the final implied volatility is shown in the final iteration.

3 Discussion

Step	Volatility	d1	d2	Fair Value	Vega
0	0.3	-0.094454	-0.17297	6.0311	16.926
1	0.43937	-0.033802	-0.1488	8.3957	16.992
2	0.43903	-0.033916	-0.14882	8.39	16.992
3	0.43903	-0.033916	-0.14882	8.39	16.992

Figure 1: Output table

This is an example using an initial guess of 0.3 as the implied volatility. After each iteration, it is adjusted based on the output of the function (the absolute value of the theoretical price minus the actual price), divided by the Vega value (the derivative of the function). When the volatility does not change by larger than 0.001 between iterations, the estimation is generated.

Initial guess (σ)	Number of iterations for convergence
0.1	3
0.2	3
0.3	3
0.4	3
0.5	2
0.6	2
0.7	3
0.8	3
0.9	3
1.0	3

The function converges very fast (2-3 iterations), suggesting that the Newton-Raphson method is likely an efficient way of determining the implied volatility of a call option. Within 3 iterations, an approximation for volatility, with precision of 10^{-3}, is generated.

4 Conclusion

In conclusion, the Newton-Raphson method has demonstrated effectiveness in determining the implied volatility within option pricing. By using a Black-Scholes model and a well-selected initial guess, the Newton-Raphson formula will iteratively approach the actual implied volatility under the function $f(x) = Call_{BS}(\sigma_{implied}) - P$. Considering implied volatility allows traders and investors to recognize the potential risk and make more informed decisions.

5 References

[1] Hayes, A. (2023, October 31). *Black-Scholes Model: What it is, how it works, options formula.* Investopedia. `https://www.investopedia.com/terms/b/blackscholes.asp`

[2] Louis, & Louis. (2019, October 29). *The Black Scholes model explained — Trade options with me.* TradeOptionsWithMe. `https://tradeoptionswithme.com/black-scholes-explained/`

[3] NEDL. (2023, April 27). *Implied volatility explained: Solver and Newton-Raphson (Excel) [Video].* YouTube. `https://www.youtube.com/watch?v=j9DrTHazR0w`

[4] Hargrave, M. (2022, March 15). *Options Contract: What it is, how it works, types of contracts.* Investopedia. `https://www.investopedia.com/terms/o/optionscontract.asp`

[5] Ganti, A. (2023, September 29). *How implied volatility (IV) works with options and examples.* Investopedia. `https://www.investopedia.com/terms/i/iv.asp`