

Straddle_pricing

June 15, 2024

1 Introduction to Quantitative Finance and Financial Risk

1.0.1 Pricing an American Straddle exercise

1.0.2 Theodoros Anagnos

1.0.3 Report on Pricing an American and European Straddle

Introduction A straddle is an options strategy that involves buying both a call and a put option with the same strike price and expiration date. Here we price an American-style at-the-money straddle using a binomial tree model and a European-style at-the-money straddle using a Monte Carlo simulation. Additionally, we discuss the delta and vega sensitivities of the American straddle.

Parameters and Assumptions

- **Current Stock Price (S_0):** 20USD
- **Strike Price (K):** 20USD
- **Time to Expiry (T):** 6 months (0.5 years)
- **Risk-Free Rate (r):** 3%
- **Dividend Yield (q):** 1.5%
- **Volatility (σ):** 35%
- **Number of Steps in Binomial Model (N):** 100
- **Number of Simulations in Monte Carlo (simulations):** 10,000

Methodology

American Straddle Pricing using Binomial Tree

1. Binomial Tree Construction:

- We set up the binomial price tree for the underlying stock. The stock price at each node is calculated based on the up and down factors (u and d) derived from the volatility and time step.
- The probability of moving up in the tree is calculated using the risk-neutral probability formula.

2. Option Valuation:

- We initialize the option value tree by calculating the payoff at expiration for both the call and the put options.
- We then work backwards through the tree, calculating the option value at each node by considering the possibility of early exercise (since these are American options).

```

[3]: import numpy as np

def american_straddle_binomial(S0, K, T, r, q, sigma, N):
    dt = T / N
    u = np.exp(sigma * np.sqrt(dt))
    d = 1 / u
    p = (np.exp((r - q) * dt) - d) / (u - d)

    # Initialize asset price tree
    stock_price = np.zeros((N + 1, N + 1))
    stock_price[0, 0] = S0
    for i in range(1, N + 1):
        stock_price[i, 0] = stock_price[i - 1, 0] * u
        for j in range(1, i + 1):
            stock_price[i, j] = stock_price[i - 1, j - 1] * d

    # Initialize option value tree
    call = np.zeros((N + 1, N + 1))
    put = np.zeros((N + 1, N + 1))

    # Calculate the option value at each node
    for j in range(N + 1):
        call[N, j] = max(0, stock_price[N, j] - K)
        put[N, j] = max(0, K - stock_price[N, j])

    for i in range(N - 1, -1, -1):
        for j in range(i + 1):
            call[i, j] = max(stock_price[i, j] - K, np.exp(-r * dt) * (p *
↪call[i + 1, j] + (1 - p) * call[i + 1, j + 1]))
            put[i, j] = max(K - stock_price[i, j], np.exp(-r * dt) * (p * put[i,
↪+ 1, j] + (1 - p) * put[i + 1, j + 1]))

    # The price of the straddle is the sum of the call and put prices at the
    ↪root
    straddle_price = call[0, 0] + put[0, 0]
    return straddle_price

S0 = 20
K = 20
T = 0.5
r = 0.03
q = 0.015
sigma = 0.35
N = 100 # Number of steps in the binomial model

american_straddle_price = american_straddle_binomial(S0, K, T, r, q, sigma, N)
print("American Straddle Price:", american_straddle_price)

```

American Straddle Price: 3.902321075200182

European Straddle Pricing using Monte Carlo Simulation

1. Simulating Stock Price Paths:

- We simulate a large number of paths for the stock price at expiration using the geometric Brownian motion formula.

2. Calculating Payoffs:

- For each simulated path, we calculate the payoff of the straddle as the sum of the payoffs from the call and put options.

3. Discounting to Present Value:

- The average payoff is discounted back to the present value to obtain the price of the straddle.

```
[4]: # Part 2: Monte Carlo Simulation for European Straddle

def european_straddle_monte_carlo(S0, K, T, r, q, sigma, simulations):
    np.random.seed(0)
    ST = S0 * np.exp((r - q - 0.5 * sigma**2) * T + sigma * np.sqrt(T) * np.
    random.standard_normal(simulations))
    payoffs = np.maximum(ST - K, 0) + np.maximum(K - ST, 0)
    price = np.exp(-r * T) * np.mean(payoffs)
    return price

simulations = 10000
european_straddle_price = european_straddle_monte_carlo(S0, K, T, r, q, sigma,
    simulations)
print("European Straddle Price:", european_straddle_price)
```

European Straddle Price: 3.836442340509254

1.0.4 Commentary on the findings:

1. American vs. European Straddle Price:

- The American straddle is slightly more expensive than the European straddle. This is expected because the American options offer more flexibility with the ability to exercise at any time before expiration, which adds value.

Sensitivity Analysis

- **Delta:** Measures the sensitivity of the option price to a small change in the underlying stock price.
- **Vega:** Measures the sensitivity of the option price to a small change in volatility.

These sensitivities were calculated by adjusting the stock price and volatility slightly and observing the changes in the straddle price.

```
[5]: # Delta and Vega calculations
def delta_vega(S0, K, T, r, q, sigma, N, epsilon=0.01):
    price = american_straddle_binomial(S0, K, T, r, q, sigma, N)
```

```

price_up = american_straddle_binomial(S0 + epsilon, K, T, r, q, sigma, N)
price_down = american_straddle_binomial(S0 - epsilon, K, T, r, q, sigma, N)
delta = (price_up - price_down) / (2 * epsilon)

price_vega_up = american_straddle_binomial(S0, K, T, r, q, sigma + epsilon,
↪N)
vega = (price_vega_up - price) / epsilon

return delta, vega

delta, vega = delta_vega(S0, K, T, r, q, sigma, N)
print("Delta:", delta)
print("Vega:", vega)

```

Delta: 0.10519698766249252

Vega: 11.04692479902516

2. Delta:

- A delta of 0.105 indicates that for a small increase in the stock price, the price of the American straddle will increase by approximately \$0.105. This relatively low delta suggests that the straddle is not highly sensitive to changes in the stock price.

3. Vega:

- A vega of 11.047 indicates that for a 1% increase in volatility, the price of the American straddle will increase by approximately \$11.05. This high vega suggests that the straddle is highly sensitive to changes in volatility, which is typical for straddle strategies as they benefit from increased volatility.

Conclusion The binomial tree and Monte Carlo methods provided effective ways to price American and European straddles respectively. The higher price of the American straddle reflects the added value of early exercise flexibility. The sensitivity analysis showed that the straddle is more responsive to changes in volatility than to changes in the underlying stock price, aligning with the characteristics of a straddle strategy that profits from significant price movements regardless of direction.