Frankfurt School

Section: Monday 1
Group 10

TOPIC 2:
PROTECTING HEDGE FUNDS FROM MARKET SQUEEZE RISK

Presentation by : Abdul Malik, Aditya Kalyanaraman, Elliott Meuer, Satya Priya Pudi, Vamsi Panchakarla

CONTENT

- Problem Statement
- Proposed Solution
- Product Design
- Replicating the Product
- Pricing the product
- Simulations: Protected vs Unprotected Outcomes
- Group Approach to Solution

PROBLEM STATEMENT

The product provides an effective risk mitigation mechanism and prevents substancial losses

Context:
Hedge funds often hold short positions to profit from anticipated declines in stock prices. Prolonged short positions can expose these funds to the risk of a short squeeze.

What is a Short Squeeze?
A short squeeze is a situation that occurs when demand increases sharply relative to supply. This can happen due to coordinated buying pressure or other market dynamics.A short squeeze can lead to substantial losses for short sellers, as they may need to buy back stocks at higher prices to cover their positions.

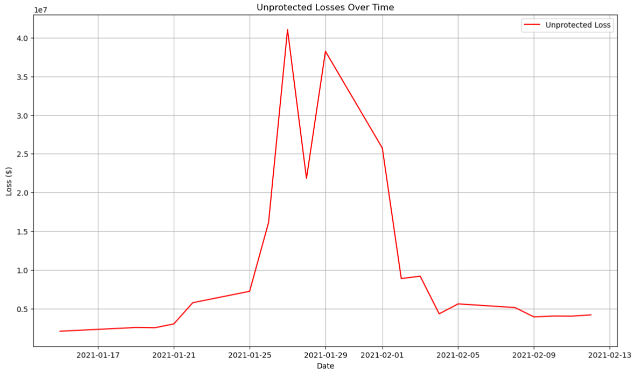
Objective:
Our Objective is to develop a financial product to mitigate the risk of short squeeze for hedgers.
This includes designing, pricing, and replicating the product to ensure robust protection while minimizing costs.

PROPOSED SOLUTION

PROPOSED SOLUTION

Prolonged short squeeze protection plan against sharp price increases

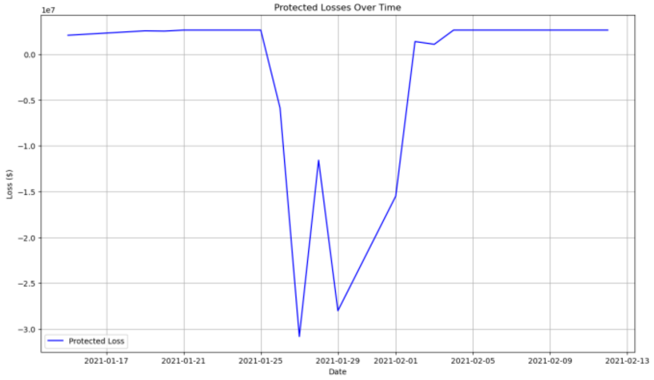
Losses Without the Hedge



PRODUCT

- Cash Settled Options
- Dynamic Hedge Management
- Liquidity Reserve

Losses/Gains With the Hedge



PRODUCT DESIGN

Our Product uses a primary hedge with dynamic hedge adjustment and a secondary Liquidity Backup. The risk is diversified by using a mix of strike prices and expiration dates to prevent dependence on a single option.

PRIMARY HEDGE

Cash-settled call options serve as the primary instrument to cap potential losses in case of price surges.
These options are customized with multiple:

- **Strike Prices:** Slightly above the current stock price for affordability.
- **Expiration Dates:** Staggered expirations aligned with potential short-squeeze events

DYNAMIC HEDGE ADJUSTMENT:
The primary hedge allocates a portion of the portfolio to a Delta-Hedging strategy to adjust exposure using liquid instruments such as ETFs or futures

SECONDARY HEDGE

A reserve fund is built into the product to address scenarios relating to liquidity constraints during options settlement. This ensures we have a reserve pool for margin calls, preventing forced liquidations.

REPLICATING THE PRODUCT

To Replicate, we use a portfolio of call options, delta hedging overlay and liquidity reserves

1. Options Portfolio

Pricing the Options:

Use the Black-Scholes model or binomial tree models to price the options. The inputs include:

- Current stock price (S_0): The market price of the stock.
- Strike price (K): Slightly above the current stock price (e.g., 5–10% higher).
- Time to maturity (T): Based on the expected duration of the short position.
- Implied volatility (σ): Volatility of the underlying stock.
- Risk-free rate (r): Yield of a zero-risk bond.

For the purpose of this demonstration, we have used Black-Scholes Model

$$C = N(d_1)S_t - N(d_2)Ke^{-rt}$$

$$\text{where } d_1 = \frac{\ln \frac{S_t}{K} + (r + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}}$$

$$\text{and } d_2 = d_1 - \sigma\sqrt{t}$$

REPLICATING THE PRODUCT

2. Dynamic Hedging

- We use liquid instruments like index futures or ETFs to adjust this component of the hedge dynamically.
- We rebalance positions daily or weekly based on changes in the underlying stock price.

3. Liquidity Reserve

Reserve Contribution:

- Collect a small fee (e.g., 1–2% of the portfolio value) to create a reserve pool for extreme liquidity scenarios.
- The reserve covers situations where options cannot be sold or exercised due to market stress.

Allocation:

- Allocate 15% of the premiums collected to the reserve.

REPLICATING THE PRODUCT

Now we create a portfolio and allocate weights to components

Constructing the Portfolio

For this demonstration we take an example of a hedge fund holding a $100M short position in a stock priced at $200. We allocate as per below breakdown:

- 35% to call options with a $210 strike price, expiring in 6 months.
- 40% to call options with a $220 strike price, expiring in 9 months.
- 25% to call options with $230 strike price expiring in 12 months .

Note : This allocation can and must be adjusted based on the volatility and liquidity of the underlying. These values are just for the sake of this example and are arbitrary.

PRICING

PRICING THE PRODUCT

Components of Pricing:

- Options Premiums: Cost of the options based on the Black-Scholes model discussed above.
- Reserve Contribution: 1- 2% of the portfolio value to fund the liquidity reserve. For this example, the liquidity reserve is 2%.
- Management Fee: A percentage of the portfolio (e.g., 0.5–1%) for active monitoring and rebalancing. 1% for this example.

Example Calculation:

- Hedge fund short position: $100M in a stock priced at $200.
- Assume the stock's volatility (σ) = 40%, risk-free rate (r) = 5%, and 6-month maturity (T = 0.5 years).
- Using Black-Scholes, calculate the premium for a $210-strike call, $220-strike call and $230 strike call.
- Assign weightage as per portfolio and calculate the average price per unit.
- Add the reserve contribution and Management Fee.

PRICING THE PRODUCT

Given the constraints in the slide above:

- Option price for strike price 210 : $19.94
- Option price for strike price 220 : $16.58
- Options price for strike price 230 : $24.73
- Average price for Option : ~ $20
- For 500,000 shares, the cost is: $20×500,000=$10M

Total cost to the client:

- Option premiums: $10M (calculated above)
- Reserve contribution: $2M (2% of notional)
- Management fee: $1M (1% of notional)
- Total: $13M

SIMULATION

Hypothetical example and real-world scenario

SIMULATION: SCENARIO COMPARISON

The product provides an effective risk mitigation mechanism and prevents substancial losses

Let's assume that due to a Short squeeze price rises from $200 to $400

UNPROTECTED LOSS

Loss **without** the product:

($400 - $200) × 500,000 Stocks

= $100M

PROTECTED LOSS

Payout

($400 - $220) × 500,000 Stocks

= $90M (Averaged for convinience)

Net Loss

$100M (unprotected loss) - $90M (payout)

= $10M

+

Product costs

= $13M

=

Protected Loss:

$13M (product costs) + $10M (residual loss)

= $23M

Savings of $77M

GAMESTOP SHORT SQUEEZE

What Happened:

In January 2021, GameStop's stock price soared due to a short squeeze, driven by retail investors on Reddit's WallStreetBets. The stock price jumped from $5 to an all-time high of $400+.Challenges for Investors: Hedge funds and short sellers faced massive losses, as they had bet on the stock falling.

We simulated the Gamestop short squeeze using our product and modeled the outcome.

The simulation outlines:

- Unprotected Loss: Losses without any options (based on stock price changes).
- Protected Loss: Losses after considering the hedge payouts from call options.
- Hedge Payout: The payout from the call options when the stock price exceeds the strike price.

Note: We have used historical daily closing price from yahoo finance to model the squeeze (due to unavailability of more frequent data)

GAMESTOP SHORT SQUEEZE

GAMESTOP SHORT SQUEEZE SIMULATION

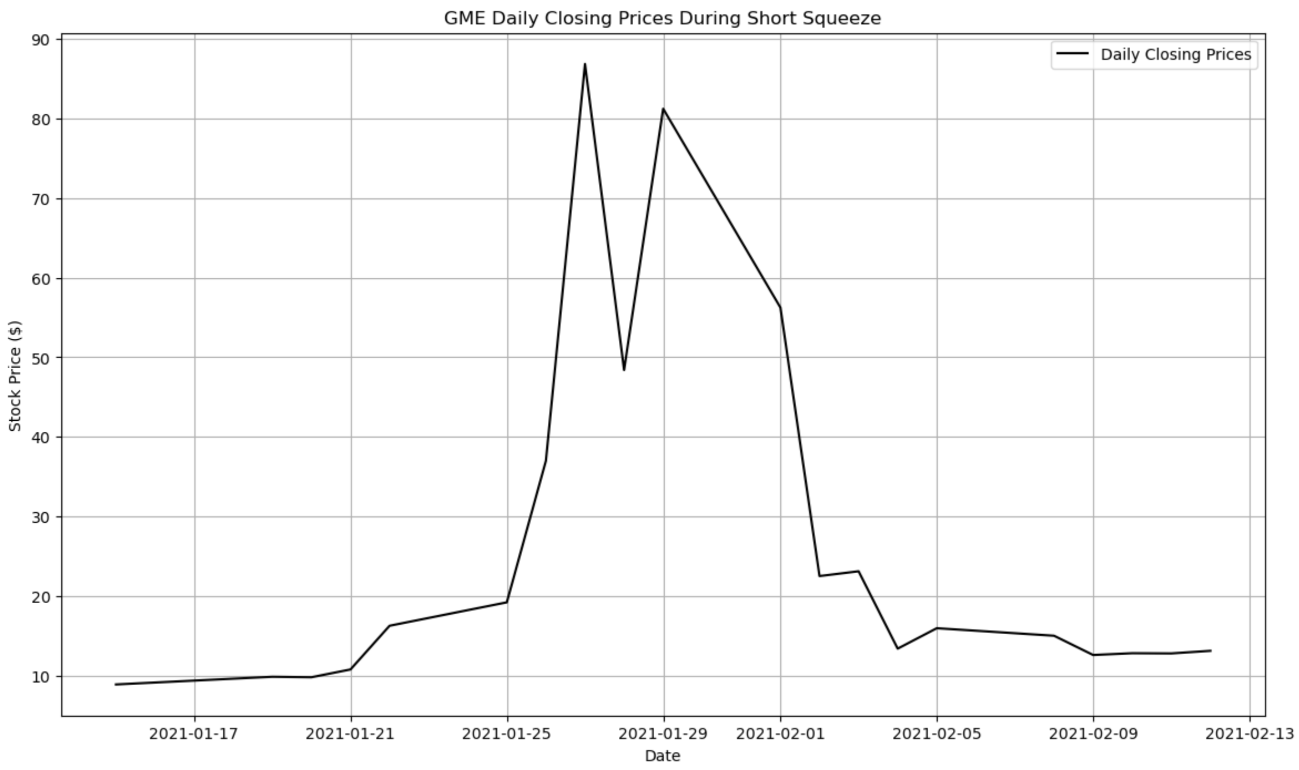
```python
# Black-Scholes Option Pricing Model
def black_scholes_call_price(S, K, T, r, sigma):
    d1 = (np.log(S / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    call_price = S * stats.norm.cdf(d1) - K * np.exp(-r * T) * stats.norm.cdf(d2)
    return call_price
```

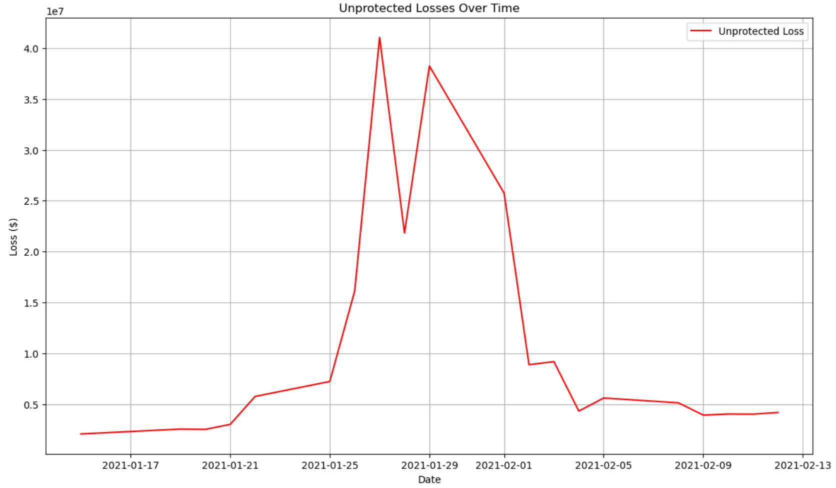
```python
# Simulate market scenarios and evaluate the performance of the protection plan
def simulate_scenario(initial_stock_price, strike_prices, maturities, volatility,
                      risk_free_rate, num_shares, final_stock_prices, liquidity_factor):
    results = []
    for final_price in final_stock_prices:
        # Calculate unprotected loss
        unprotected_loss = (final_price - initial_stock_price) * num_shares

        # Calculate protected loss
        hedge_payouts = []
        for K, T in zip(strike_prices, maturities):
            payout = max(0, final_price - K) * num_shares
            hedge_payouts.append(payout)

        total_hedge_payout = sum(hedge_payouts)
        protected_loss = unprotected_loss - total_hedge_payout

        results.append({
            "Final Stock Price": final_price,
            "Unprotected Loss": unprotected_loss,
            "Protected Loss": protected_loss,
            "Hedge Payout": total_hedge_payout,
        })
    return results
```

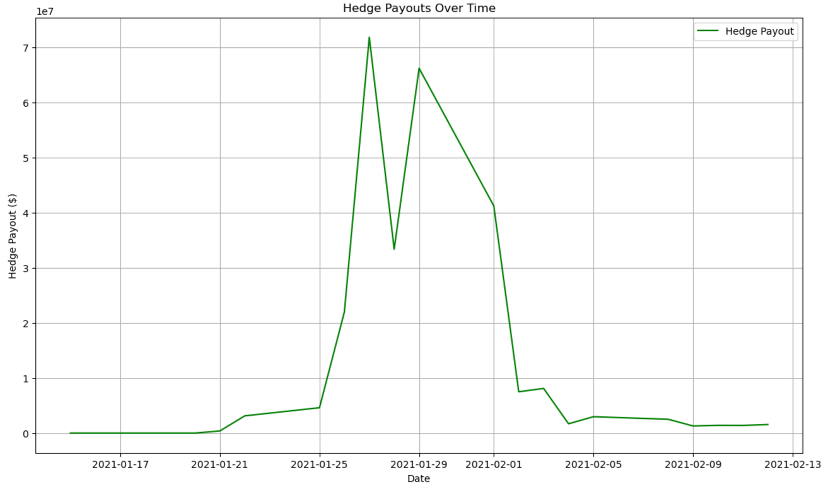
```python
# Parameters
initial_stock_price = 4.75
strike_prices = [10, 20]
maturities = [0.25, .5]  # in years
volatility = .5
risk_free_rate = 0.05
num_shares = 500000
ticker = "GME"
start_date = "2021-01-01"
end_date = "2021-03-01"
liquidity_factor = 1.1
```

Detailed code with graphs attached in the appendix.

GAMESTOP SHORT SQUEEZE SIMULATION







Total Cost of Protection Plan: $0.07M for 500,000 shares of Gamestop purchased at $4.75/share ($2.375M)

GROUP APPROACH AND LLM INTERACTION

Diverse roles: Our team members bring unique skills

- We decided to take this topic as a case study of some recent short squeezes. During the whole process we discussed short squeezes starting from VW squeeze of 2008, Herbalife, AMC and Gamestop.
- Gamestop being the most recent and publicised, we decided to use this for our modelling.
- We started off by brainstorming and critiquing ideas. Once we had a general understanding, we posed directed questions to ChatGPT and then refined our interaction by pointing out errors in its approach.
- Once we were satisied with the approach, we started testing with snippets of code and refining our assumptions and pricing metrics.
- Lastly, we tested our code with data from yfinance to model a viable product.

TEAM 10

Abdul Malik

Aditya Kalyanaraman

Elliott Meuer

Satya Priya Pudi

Vamsi Panchakarla

THANK YOU!

APPENDIX

~\Downloads\FPM\Project.ipynb

```python
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import yfinance as yf
import pandas as pd

# Black-Scholes Option Pricing Model
def black_scholes_call_price(S, K, T, r, sigma):
    d1 = (np.log(S / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    call_price = S * stats.norm.cdf(d1) - K * np.exp(-r * T) * stats.norm.cdf(d2)
    return call_price

# Adjust for liquidity
def adjust_for_liquidity(premium, liquidity_factor):
    return premium * liquidity_factor

# Simulate market scenarios and evaluate the performance of the protection plan
def simulate_scenario(initial_stock_price, strike_prices, maturities, volatility,
risk_free_rate, num_shares, final_stock_prices, liquidity_factor):
    results = []
    for final_price in final_stock_prices:
        # Calculate unprotected loss
        unprotected_loss = (final_price - initial_stock_price) * num_shares

        # Calculate protected loss
        hedge_payouts = []
        for K, T in zip(strike_prices, maturities):
            payout = max(0, final_price - K) * num_shares
            hedge_payouts.append(payout)

        total_hedge_payout = sum(hedge_payouts)
        protected_loss = unprotected_loss - total_hedge_payout

        results.append({
            "Final Stock Price": final_price,
            "Unprotected Loss": unprotected_loss,
            "Protected Loss": protected_loss,
            "Hedge Payout": total_hedge_payout,
        })
    return results

# Fetch historical stock prices during the GameStop short squeeze
def fetch_historical_prices(ticker, start_date, end_date):
    stock_data = yf.download(ticker, start=start_date, end=end_date)
    return stock_data
```

```python

# Parameters
initial_stock_price = 4.75
strike_prices = [10, 20]
maturities = [0.25, .5]  # in years
volatility = .5
risk_free_rate = 0.05
num_shares = 500000
ticker = "GME"
start_date = "2021-01-01"
end_date = "2021-03-01"
liquidity_factor = 1.1

# Fetch real-world stock prices
stock_data = fetch_historical_prices(ticker, start_date, end_date)
final_stock_prices = stock_data['Close'].values
dates = stock_data.index

# Simulate and display results
results = simulate_scenario(initial_stock_price, strike_prices, maturities,
volatility, risk_free_rate, num_shares, final_stock_prices, liquidity_factor)

# Extract data for plotting
final_prices = [result["Final Stock Price"] for result in results]
unprotected_losses = [result["Unprotected Loss"] for result in results]
protected_losses = [result["Protected Loss"] for result in results]
hedge_payouts = [result["Hedge Payout"] for result in results]

# Plot the daily closing prices during the short squeeze period
plt.figure(figsize=(14, 8))
plt.plot(dates, final_stock_prices, label='Daily Closing Prices', color='black')
plt.xlabel('Date')
plt.ylabel('Stock Price ($)')
plt.title('GME Daily Closing Prices During Short Squeeze')
plt.legend()
plt.grid(True)
plt.show()

# Plot the results: Unprotected vs. Protected Losses
plt.figure(figsize=(14, 8))
plt.plot(dates, unprotected_losses, label='Unprotected Loss', color='red')
plt.plot(dates, protected_losses, label='Protected Loss', color='blue')
plt.xlabel('Date')
plt.ylabel('Loss ($)')
plt.title('Unprotected vs. Protected Losses Over Time')
plt.legend()
plt.grid(True)
plt.show()
```

```python

# Plot the results: Total Hedge Payouts Over Time
plt.figure(figsize=(14, 8))
plt.plot(dates, hedge_payouts, label='Hedge Payout', color='green')
plt.xlabel('Date')
plt.ylabel('Hedge Payout ($)')
plt.title('Hedge Payouts Over Time')
plt.legend()
plt.grid(True)
plt.show()

# Plot histogram of stock prices during the short squeeze period
plt.figure(figsize=(14, 8))
plt.hist(final_stock_prices, bins=30, color='purple', edgecolor='black')
plt.xlabel('Stock Price ($)')
plt.ylabel('Frequency')
plt.title('Histogram of GME Stock Prices During Short Squeeze')
plt.grid(True)
plt.show()

# Display a summary table of the results
summary_table = pd.DataFrame({
    'Date': dates,
    'Final Stock Price': final_prices,
    'Unprotected Loss': unprotected_losses,
    'Protected Loss': protected_losses,
    'Hedge Payout': hedge_payouts
})
print(summary_table)

# Calculate total cost of the protection plan
def calculate_total_cost(initial_stock_price, strike_prices, maturities, volatility, risk_free_rate, num_shares, management_fee_rate, reserve_rate, liquidity_factor):
    option_premiums = []
    for K, T in zip(strike_prices, maturities):
        call_price = black_scholes_call_price(initial_stock_price, K, T, risk_free_rate, volatility)
        adjusted_call_price = adjust_for_liquidity(call_price, liquidity_factor)
        total_call_price = adjusted_call_price * num_shares
        option_premiums.append(total_call_price)

    total_option_premiums = sum(option_premiums)
    reserve_contribution = reserve_rate * initial_stock_price * num_shares
    management_fee = management_fee_rate * initial_stock_price * num_shares

    total_cost = total_option_premiums + reserve_contribution + management_fee
    return total_cost

# Parameters for cost calculation
```

```
management_fee_rate = 0.01
reserve_rate = 0.02

total_cost = calculate_total_cost(initial_stock_price, strike_prices, maturities, volatility, risk_free_rate, num_shares, management_fee_rate, reserve_rate, liquidity_factor)
print(f"Total Cost of Protection Plan: ${total_cost / 1e6:.2f}M")

# Check for negative protected losses and provide an explanation
for result in results:
    if result["Protected Loss"] < 0:
        print(f"Date: {dates[results.index(result)]}, Final Stock Price: {result['Final Stock Price']}, Net Gain: ${-result['Protected Loss']:.2f}")
```