Goldman Sachs India Hackathon 2025

Quant Presentation

Soumyadeb Saha

IIT Kanpur

24 June, 2025

1. Optimal Hedging Strategy

Objective: The goal is to strategically select stocks and their quantities to minimize the provided unhedged losses, while also minimizing the total cost hedging which is associated with each stock.

Minimizing Hedged P&L $\mathcal{L}_1(\mathbf{w}) = \mathrm{Var}(\mathbf{y} + \mathbf{X}\mathbf{w})$

Minimizing Cost $\mathcal{L}_2(\mathbf{w}) = \sum_{i=1}^{N} |w_i| \cdot c_i$

$\mathbf{y} \in \mathbb{R}^T$: vector of **unhedged P&L** over T days.

$\mathbf{X} \in \mathbb{R}^{T \times N}$: matrix of **daily returns** for N stocks.

$\mathbf{w} \in \mathbb{R}^N$: vector of **weights/quantities** of stocks

$\mathbf{c} \in \mathbb{R}^N$: vector of **capital costs** for each stock.

Approach:

- Stock returns were scaled by dividing each return by its corresponding hedging cost (**Scaled Returns**) to account for implementation expenses.
- **Lasso** regression was applied to these scaled returns to estimate coefficients that minimize the portfolio's risk while considering costs.
- The obtained coefficients were then rescaled by dividing by the hedging costs to convert them back into actual hedge quantities.
- A minimum absolute threshold of 10^{-6} units was set on the hedge quantities to exclude insignificant positions.

Limitations

- The approach is not considering the Cost Minimization loss function effectively.
- The quantities generated without considering total cost has significant similarity.

A key drawback was that the model performed well on all visible test cases even without incorporating capital costs

```
stock_125 -1389
stock_130 14
```

without Scaled Returns

```
stock_125 -1283
stock_141 16
stock_185 -14
```

with Scaled Returns

Better Approach:

The strategy minimizes a combined score of variance and cost by tuning two parameters: **regularization (α)** and **trade-off (λ).**

$$L(w) = \frac{1}{2n}\|y - Xw\|_2^2 + \alpha \sum_{i=1}^{N} |w_i|$$

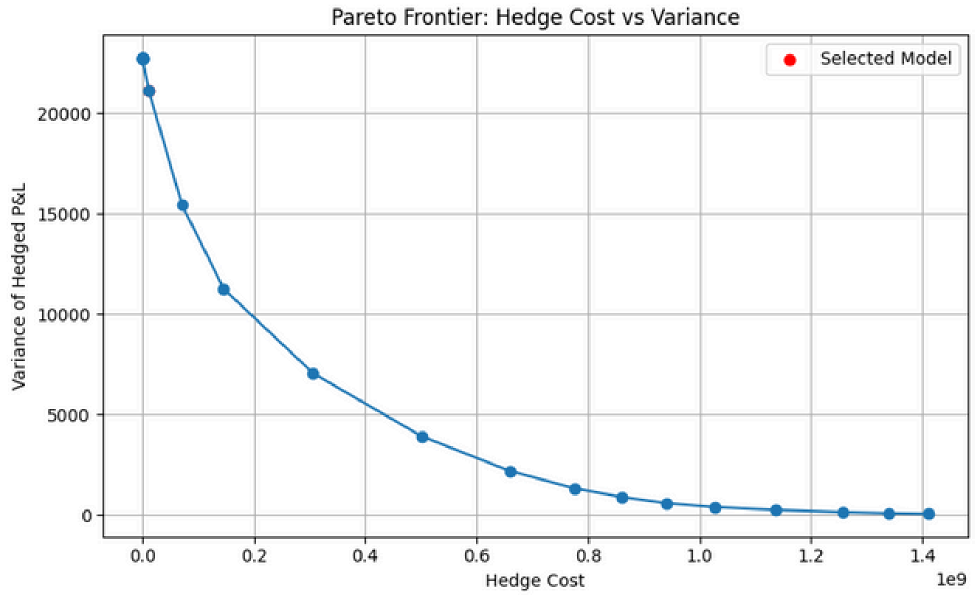
$$\mathrm{Score}(\mathbf{w}) = \underbrace{\mathrm{Var}\left(\mathbf{y} + X\mathbf{w}\right)}_{\text{Hedged PnL Variance}} + \lambda \cdot \underbrace{\sum_{i=1}^{N} |w_i| \cdot c_i}_{\text{Total Hedging Cost}}$$

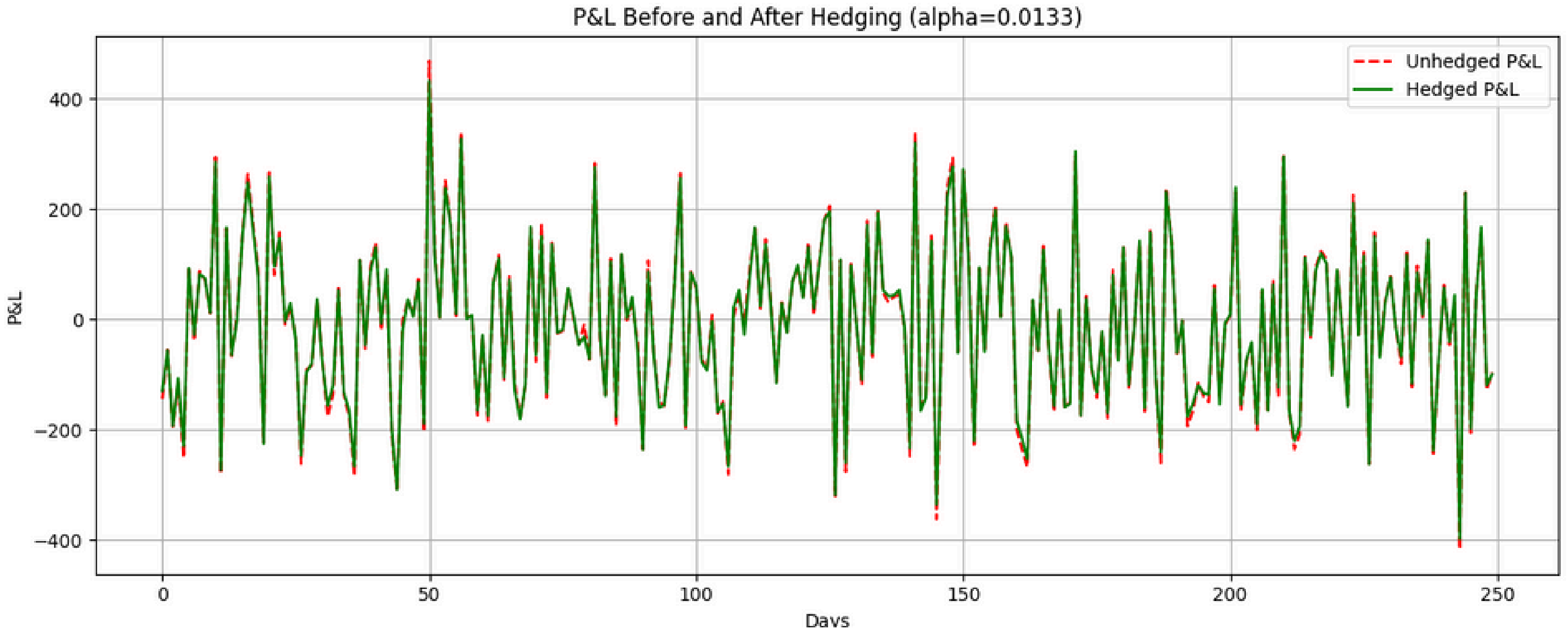
- Regularization parameter **α** is varied from **10^{-4}** to **10^{4}** in **50 intervals** to get coeff.
- For each **(α), trade-off (λ)** is varied from **10^{-1} to 10^{-4}** in **30** intervals. to get **Score**
- The {**α ,λ** } pair giving the lowest score across all combinations is selected.

Improved Result:

Lowest Loss Score is obtained at: **α = 0.0133 & λ = 0.001**

Improved the overall result **88.24 → 91.18**





2. Automated Market Making

Volatility spread based Strategy :

- Defined **mid-price** as the average of the maximum bid and minimum ask prices
- Implemented the **Avellaneda-Stoikov model** with **inventory-aware quoting logic**.
- Used a rolling window of **10 timestamps** to dynamically update the reservation price.
- In **high-volatility periods**, the model widens the spread by raising the ask price and lowering the bid price to limit risk and manage exposure

$m = \text{mid-price},\ \sigma = \text{volatility},\ T = \text{horizon}$ $\gamma = \text{risk aversion},\ q = \text{inventory},\ \delta = \text{tick size}$	$r = m - q \cdot \gamma \cdot (2\sigma)^2 \cdot T$ $s = \delta + 2 \cdot \gamma \cdot (2\sigma)^2 \cdot T$	$\text{bid} = r - \frac{s}{2}, \quad \text{ask} = r + \frac{s}{2}$

Aggressive Inventory Management Strategy :

- **When inventory exceeds +20 :** The strategy increases the bid price to attract more buyers and create incentive for them to take the asset.
- **When inventory reduces below -20:** The strategy decreases the bid price to reduce selling pressure and prevent further depletion of the asset.

For Positive Inventory (Inventory ≥ 20):

- $\text{bid\_price} = \max(\text{bid}) - 2 \cdot \delta$
- $\text{ask\_price} = \min\left(\min(\text{ask}) - \delta,\ \text{bid\_price} + \delta\right)$

For Negative Inventory (Inventory ≤ -20):

- $\text{ask\_price} = \min(\text{ask}) + 2 \cdot \delta$
- $\text{bid\_price} = \max\left(\max(\text{bid}) + \delta,\ \text{ask\_price} - \delta\right)$

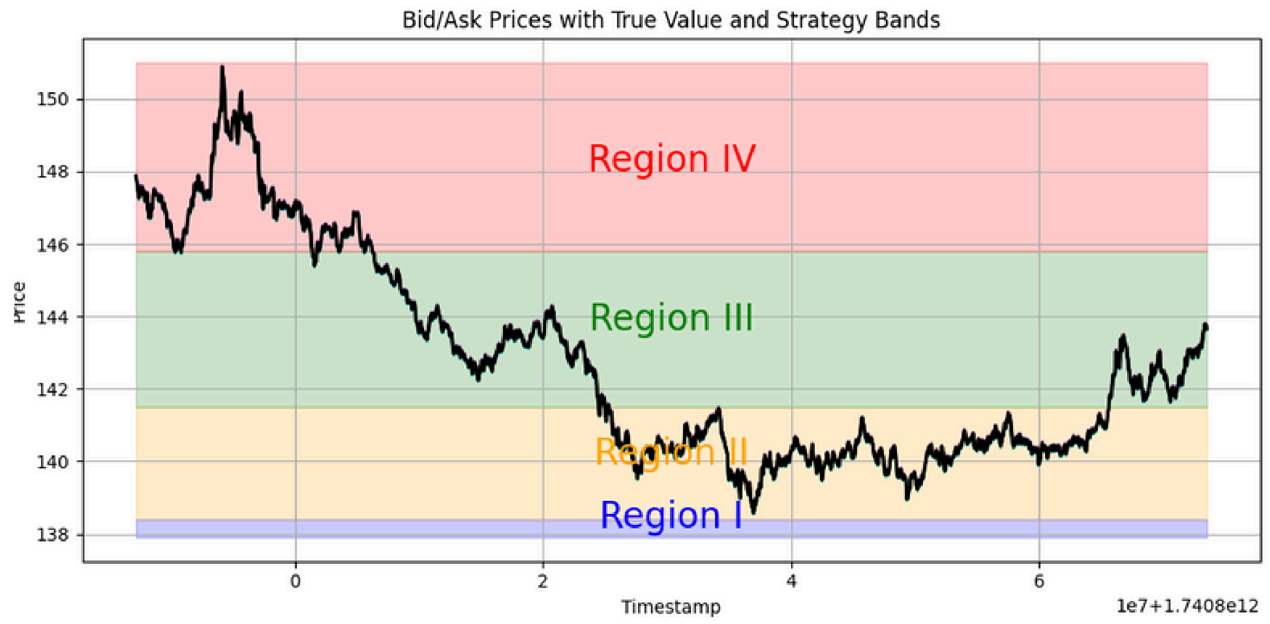
Mid-price based Quotes:

Region -I : Strong Buy Zone
Bid price → Bid Price + 0.1,
Ask Price → Ask Price + 0.5

Region -II : Buy Bias Zone
Bid price → Bid Price
Ask Price → Ask Price + 0.3

Region -III : Neutral Zone
Bid price → Bid Price - 0.3,
Ask Price → Ask Price + 0.3

Region -IV: Sell Bias Zone
Bid price → Bid Price - 0.5
Ask Price → Ask Price



Limitations :

- The strategy fails to market-make when the midprice moves outside the order book range (lowest bid -- highest ask)
- It does not utilized bid and ask prices from the **public trades**

Alternate Robust Strategy:

Momentum-Based Price Adjustment :

- When the mid-price shows a consistent trend (increasing or decreasing) over a 30-timestamp window, the algorithm interprets this as directional market movement.
- In response, a directional price adjustment is applied to gradually reduce inventory levels, aligning with the price trend.
- The bid and ask prices are adjusted slightly to encourage trades that move the inventory toward zero.

$q = \text{inventory}$
$p_{\text{bid\_pub}} = \text{best public bid}$
$p_{\text{ask\_pub}} = \text{best public ask}$
$\delta_{\text{tick}} = \text{tick size}$

Case 1: $q \geq 16$ (excess long inventory)

$$p_{\text{bid}} = p_{\text{bid\_pub}} - 3 \cdot \delta_{\text{tick}}$$

$$p_{\text{ask}} = \min\left(p_{\text{ask\_pub}} - \delta_{\text{tick}},\ p_{\text{bid}} + \delta_{\text{tick}}\right)$$

Case 2: $q \leq -16$ (excess short inventory)

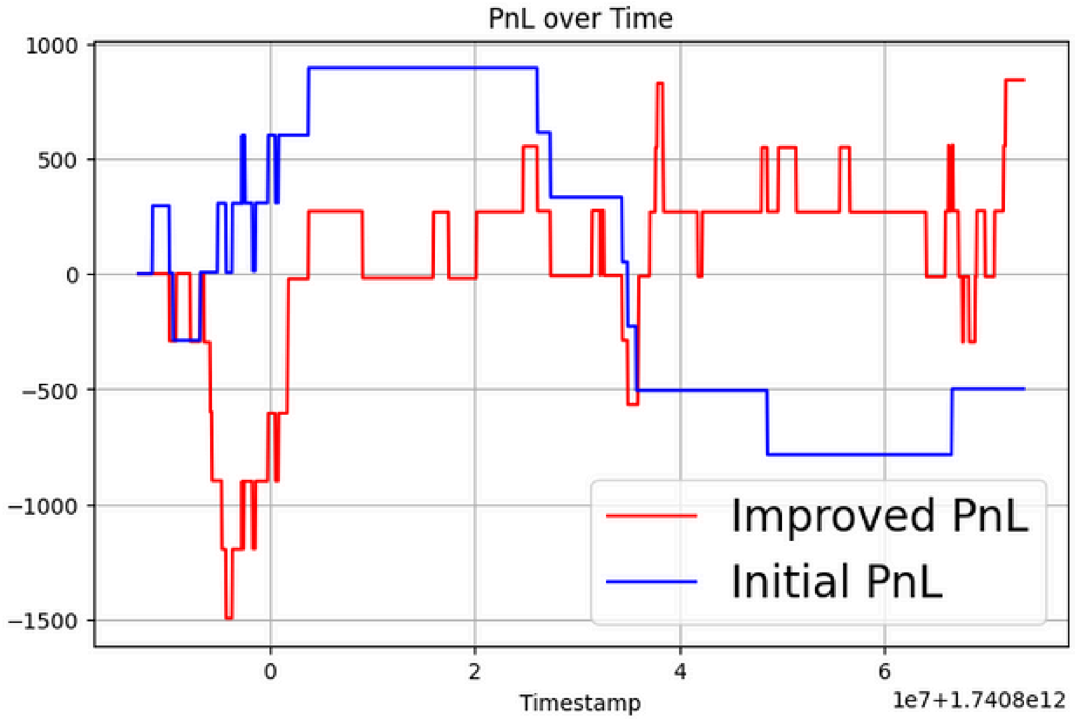
$$p_{\text{ask}} = p_{\text{ask\_pub}} + 3 \cdot \delta_{\text{tick}}$$

$$p_{\text{bid}} = \max\left(p_{\text{bid\_pub}} + \delta_{\text{tick}},\ p_{\text{ask}} - \delta_{\text{tick}}\right)$$

Improved PnL :

In The backtester :

After incorporating a **Momentum**-driven quoting strategy in place of the traditional mid-price approach, the overall PnL exhibited a notable enhancement,



3 . Exotic Option Pricing using MonteCarlo :

Objective :

- **Aim is to obtain price of Knock Out Put and Call , using Strike Price , Barrier and Maturity**
- **Additionally we need to generate Spot Price(t) using and caliberate Local Volatility using MonteCarlo simulation**

Approach:

Using **Black Scholes** Implied Volatility formula and **Brent** Root Finding method calculated **IV** for each stocks at a fixed Strike price **K** and Maturity **T**

→

Forming Implied Volatility Matrix

for each stock $s \in \{\mathrm{DTC}, \mathrm{DFC}, \mathrm{DEC}\}$

$$\mathbf{V}^{(s)} = \begin{bmatrix} \mathrm{IV}_{1,1}^{(s)} & \mathrm{IV}_{1,2}^{(s)} & \mathrm{IV}_{1,3}^{(s)} \\ \mathrm{IV}_{2,1}^{(s)} & \mathrm{IV}_{2,2}^{(s)} & \mathrm{IV}_{2,3}^{(s)} \\ \mathrm{IV}_{3,1}^{(s)} & \mathrm{IV}_{3,2}^{(s)} & \mathrm{IV}_{3,3}^{(s)} \\ \mathrm{IV}_{4,1}^{(s)} & \mathrm{IV}_{4,2}^{(s)} & \mathrm{IV}_{4,3}^{(s)} \\ \mathrm{IV}_{5,1}^{(s)} & \mathrm{IV}_{5,2}^{(s)} & \mathrm{IV}_{5,3}^{(s)} \end{bmatrix} \in \mathbb{R}^{5\times 3}$$

Rows i correspond to strike levels: $[50, 75, 100, 125, 150]$

Columns j correspond to maturities: $[1, 2, 5]$ (years)

→

Relation of Local Volatility and Implied Volatility

Previous Spot price is used to calculate the **Current Spot** by mapping a probable **Strike (K)** and **maturity (T)**

Step 1: Map each $S^{(s,i)} \to$ closest strike $K^{(i)} \leq S^{(s,i)}$

Step 2: Identify $T^{(t)} \in \{1, 2, 5\}$ based on current time

Step 3: Look up local vol: $\sigma^{(s,i)} = \sigma^{(s)}(K^{(i)}, T^{(t)})$

$$K_{\text{mapped}}^{(\text{stock})} = [K_1, K_2, \ldots, K_{n_{\text{path}}}] \quad \text{time\_index} = \begin{cases} 0 & \text{if } t \leq 1 \\ 1 & \text{if } 1 < t \leq 2 \\ 2 & \text{if } 2 < t \leq 5 \end{cases}$$

The local Volatility is calculated directly from Matrix using **mapped Strike (K)** and **time_index (t)**

↓

Calculation of New Spot Price

Provided **DIfferentail equation** of **Spot Price** contains both **Drift** and **Diffusion part.**

$$\text{Drift} = r \cdot S(t) \cdot dt$$

$$\mathbf{D}_{\text{diff}} = \boldsymbol{\sigma} \odot \mathbf{S}_{\text{prev}} \odot \sqrt{dt} \odot \mathbf{Z}_{\text{corr}}^{\top}$$

$$dt = \frac{1}{\text{time steps per year}}$$

Z_corr: represents randomness via Brownian motion.

To include correlated random shocks in the diffusion , standard normal vector **Z** is multiplied with the **Cholesky-decomposed correlation matrix**

$$\begin{bmatrix} Z_1^{\text{corr}} \\ Z_2^{\text{corr}} \\ Z_3^{\text{corr}} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} Z_1^{\text{uncorr}} \\ Z_2^{\text{uncorr}} \\ Z_3^{\text{uncorr}} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1.00 & 0.75 & 0.50 \\ 0.75 & 1.00 & 0.25 \\ 0.50 & 0.25 & 1.00 \end{bmatrix}$$

$$\mathbf{C} = \mathbf{L} \cdot \mathbf{L}^{\top}$$

$$\mathbf{Z}_{\text{corr}} = \mathbf{L} \cdot \mathbf{Z}_{\text{uncorr}}^{\top}$$

←

Price Calculation Using Payoff

$$\text{DTC} \Leftarrow \begin{bmatrix} S_{0,1}^{\text{DTC}} \\ S_{0,2}^{\text{DTC}} \\ \vdots \\ S_{0,n}^{\text{DTC}} \end{bmatrix} \xrightarrow{t=1} \begin{bmatrix} S_{1,1}^{\text{DTC}} \\ S_{1,2}^{\text{DTC}} \\ \vdots \\ S_{1,n}^{\text{DTC}} \end{bmatrix} \xrightarrow{t=2} \cdots \xrightarrow{t=T} \begin{bmatrix} S_{T,1}^{\text{DTC}} \\ S_{T,2}^{\text{DTC}} \\ \vdots \\ S_{T,n}^{\text{DTC}} \end{bmatrix}$$

$$\text{DFC} \Leftarrow \begin{bmatrix} S_{0,1}^{\text{DFC}} \\ S_{0,2}^{\text{DFC}} \\ \vdots \\ S_{0,n}^{\text{DFC}} \end{bmatrix} \xrightarrow{t=1} \begin{bmatrix} S_{1,1}^{\text{DFC}} \\ S_{1,2}^{\text{DFC}} \\ \vdots \\ S_{1,n}^{\text{DFC}} \end{bmatrix} \xrightarrow{t=2} \cdots \xrightarrow{t=T} \begin{bmatrix} S_{T,1}^{\text{DFC}} \\ S_{T,2}^{\text{DFC}} \\ \vdots \\ S_{T,n}^{\text{DFC}} \end{bmatrix}$$

$$\text{DEC} \Leftarrow \begin{bmatrix} S_{0,1}^{\text{DEC}} \\ S_{0,2}^{\text{DEC}} \\ \vdots \\ S_{0,n}^{\text{DEC}} \end{bmatrix} \xrightarrow{t=1} \begin{bmatrix} S_{1,1}^{\text{DEC}} \\ S_{1,2}^{\text{DEC}} \\ \vdots \\ S_{1,n}^{\text{DEC}} \end{bmatrix} \xrightarrow{t=2} \cdots \xrightarrow{t=T} \begin{bmatrix} S_{T,1}^{\text{DEC}} \\ S_{T,2}^{\text{DEC}} \\ \vdots \\ S_{T,n}^{\text{DEC}} \end{bmatrix}$$

The spot prices for each stock at each time (t)

$$\text{Option Price} = e^{-rT} \cdot \frac{1}{n_{\text{valid}}} \sum_{i=1}^{n_{\text{valid}}} \text{Payoff}_i$$

At each time calculate Basket Price

$$S_t^{\text{Basket}} = \frac{1}{3}\left(S_t^{\text{DTC}} + S_t^{\text{DFC}} + S_t^{\text{DEC}}\right)$$

If Basket Price exceed **Barrier B** that path is **removed**

$$S_t^{\text{Basket}} = \begin{bmatrix} S_{t,1}^{\text{Basket}} \\ S_{t,2}^{\text{Basket}} \\ \vdots \\ S_{t,n_{\text{valid}}}^{\text{Basket}} \end{bmatrix}$$

At Maturity T , final Payoff is calculated

$$\text{Payoff}^{\text{Call}} = \begin{bmatrix} \max(S_{T,1}^{\text{Basket}} - K, 0) \\ \max(S_{T,2}^{\text{Basket}} - K, 0) \\ \vdots \\ \max(S_{T,n_{\text{valid}}}^{\text{Basket}} - K, 0) \end{bmatrix}$$

Limitations:

In **Local volatility calculation** the **Direct mapping** of Strike prices with Implied volatility makes the path simulation unstable and noisy (due to **volatility jumps**)

Improvements

Bilinear Interpolation method reduces jumps

- Obtain **4 nearest pair** of **Strike & Maturity** using previous **SpotPrice (S)** and **time (t)** .
- Obtain the Implied Volatility of **4 pairs (K ,T)**
- The local volatility of a path will be the **weighted average of** 4 Implied volatilities.

Results & Plots :

Strike = 100 , **Barrier** = 150 , **Maturity** = 2

Option = Put , **Path**=2000 ,