Optimal Trade Execution Using Market Impact Modeling

Name: Mohamed Ashraf

Task 1: Market Impact Modeling

Step 1: Load and Preprocess LOB Data

- Use MBP-10 data from April 2025 (top 10 levels of bid/ask prices).
- Convert nanodollar prices to dollars.
- Group by minute using the last message of each minute.

Step 2: Simulate Market Buy Orders

- For each minute, simulate a market buy order of size x using the top 10 ask levels.
- Compute the **volume-weighted average price (VWAP)**:

$$\mathrm{VWAP}_t(x) = \frac{\sum p_i \cdot s_i}{x}$$

- Calculate **slippage**:

$$g_t(x) = \mathrm{VWAP}_t(x) - \mathrm{Mid}_t$$

Where:

$$\mathrm{Mid}_t = \frac{\mathrm{BestBid}_t + \mathrm{BestAsk}_t}{2}$$

Step 3: Fit Slippage Models

Two types of models are fit using `curve_fit`:

1. **Linear Model**:

$$g(x) = \beta x$$

2. **Quadratic Model**:

$$g(x) = \alpha x^2 + \beta x$$

- The quadratic model captures curvature in price impact better than the linear one.
- Fitting is done independently for each minute t.

Task 2: Optimal Execution with CVXPY

Goal

Buy a total of $S = 10,000$ shares over $T = 390$ minutes, minimizing expected execution cost.

Mathematical Formulation

Let x_t be the number of shares to buy at minute t.

Temporary impact model per minute:

$$g_t(x_t) = \alpha_t x_t^2 + \beta_t x_t$$

Objective:

$$\min_{x_1,...,x_T} \sum_{t=1}^{T} \left(\alpha_t x_t^2 + \beta_t x_t\right)$$

Subject to:

$$\sum_{t=1}^{T} x_t = S, \quad x_t \geq 0$$

- Solved using **CVXPY**, a convex optimization solver in Python.

Results

- The optimal execution schedule allocates more trades during low-slippage periods.
- Compared with a **uniform strategy** (constant $x_t = \frac{S}{T}$), the optimal solution reduces total cost.

Execution Schedule Visualization

- **Optimal Execution**: Varies per minute based on fitted α_t and β_t.
- **Uniform Execution**: Flat baseline at S/T shares per minute.