

A study on temporary impact and the Almgren-Chriss model, an optimal order execution strategy

Chaitanya Agarwal

31 July 2025

1 Modeling temporary impact $g_t(x)$

Code at github.com/Chaim3ra/modeling-temp-impact.

We know, the temporary impact function $g_t(x)$ can be defined as the amount of slippage that occurs if x orders are placed at the current time t . To effectively model $g_t(x)$, I analyzed the given MBP-10 snapshots of three companies, namely CoreWeave Inc(CRWV), JFrog Ltd(FROG), and SoundHound AI Inc(SOUN). Each snapshot provides the top ten bid and ask price-size pairs, from which I compute the mid-price m_t and the spread. For each snapshot, I then "walk" the ask book, filling shares at successive price levels until x shares are filled. Slippage can then be calculated as:

$$g_t(x) = \frac{1}{x} \sum_{k=1}^K (p_k - m_t) \Delta x_k$$

For my initial simulations, I considered the following order sizes $\{5, 50, 100, 500, 1000\}$ to get a rough sense of the shape of the slippage curves. After plotting the median slippage curves on log-log axes(Figure 1), we observe that for CRWV and FROG, slippage grows with order size. However, slippage remains flat for SOUN until larger order sizes($x \geq 1000$). To capture a more representative curve for SOUN, I used a log-spaced grid of 7 unique points from 50 to 5000 shares(which increased the filesize from 3.38GB to 4.45GB).

The updated median slippage curves(Figure 2) now show SOUN rises sharply as the order size increases past 1000. The curves exhibit a concave trend where smaller

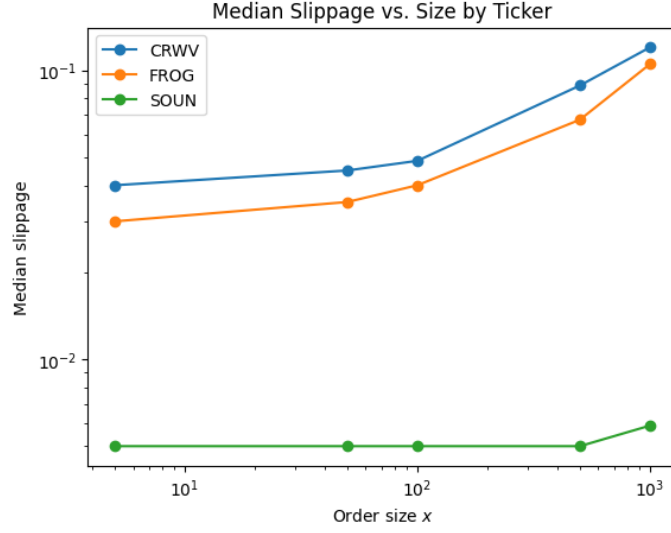


Figure 1: Median Slippage for fixed grid

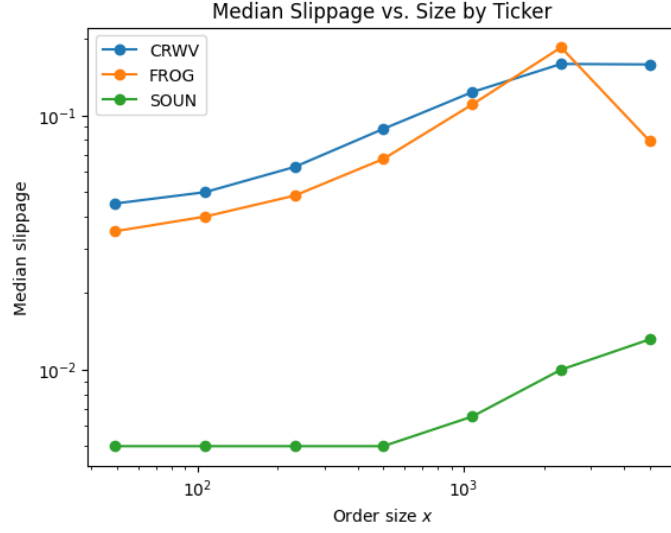


Figure 2: Median Slippage for log-spaced grid

order sizes incur a very small cost. If the order size was to be doubled, the increase in slippage would be far less than twofold. From my research, this behaviour points us towards two potential models:

- **Linear:** $g(x) = \beta x$, which will serve as my baseline.

- **Power-law:** $g(x) = \alpha x^\gamma$, which exhibits concavity similar to the slippage curves.

I fit the aforementioned models to the median slippage data and report the estimated parameters, RMSE, and R^2 for the models, proving that power-law fits significantly better than any linear model.

Ticker	β	RMSE	R^2
CRWV	4.30×10^{-5}	3.407×10^{-3}	-0.6382
FROG	3.10×10^{-5}	4.483×10^{-3}	-0.8701
SOUN	3.00×10^{-6}	1.400×10^{-5}	-0.5688

Table 1: Linear fit results.

Ticker	α	γ	RMSE	R^2
CRWV	1.2402×10^{-2}	0.3144	1.42×10^{-4}	0.9649
FROG	1.1103×10^{-2}	0.2940	1.393×10^{-3}	0.6835
SOUN	1.7750×10^{-3}	0.2112	2.00×10^{-6}	0.7733

Table 2: Power-law fit results.

Table 1 shows that the linear model fails to accurately predict slippage at small order sizes (negative R^2) and comes close only near larger order sizes. On the other hand, from Table 2 we observe that power-law achieves a very high log-log R^2 on average for the three tickers, thus proving its ability to effectively capture concave trends as observed in our slippage curves.

In the section below, I will implement the Almgren-Chriss model to formulate an optimal order executing framework to compute x_i that minimizes total expected slippage.

2 Order Execution Strategy

We know, the temporary impact model $g(x)$ tells us the slippage per-share for any given order size x . Let's suppose that we have S total amount of orders that need to be filled over N equal trading periods, we need to decide how many shares x_1, x_2, \dots, x_N to trade in each trading period so as to minimize slippage. That is:

$$\min_{x_1, \dots, x_N \geq 0} \sum_{i=1}^N g(x_i) \quad \text{subject to} \quad \sum_{i=1}^N x_i = S,$$

Thus, this is an optimization problem where each x_i represents the shares executed in trading period i , with the goal to minimize temporary impact.

To determine how to slice S over N intervals, we use the Almgren-Chriss model. In the optimal execution model, we work with:

- How the price moves
- How our trading affects prices, which is our function from the previous section $g(x)$.

The Almgren-Chriss model is given by:

$$\min_{x_1, \dots, x_N} \sum_{i=1}^N g(x_i) + \lambda \text{Var} \left[\sum_{i=1}^N x_i \tilde{p}_i \right] \quad \text{s.t.} \quad \sum_{i=1}^N x_i = S,$$

where λ is the risk-aversion parameter and \tilde{p}_i are the random price moves. If we focus only on how our trading affects prices, the problem reduces to:

$$\min_{x_1, \dots, x_N} \sum_{i=1}^N g(x_i) \quad \text{s.t.} \quad \sum_{i=1}^N x_i = S.$$

We can now fit the parameters η and Γ to the median slippage data, which can then be solved using CVXPY (a convex optimization library in Python) to calculate the optimal shares(x_i).

Another way to formulate an optimal order executing framework would be to utilize our power-law model from the previous section. That is:

$$\min \sum \alpha x_i^\gamma$$

As this is a concave function, we would need to minimize it using a non-convex solver such as DCCP.

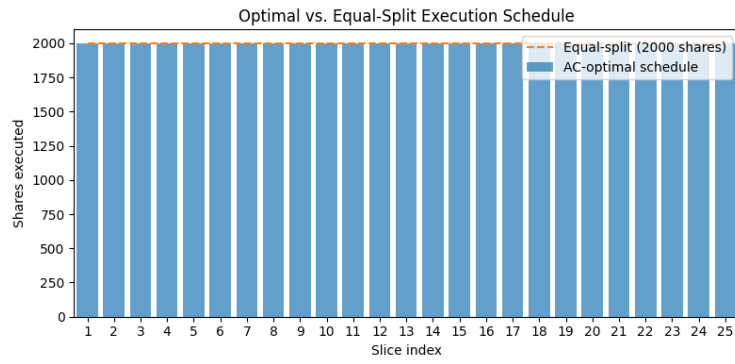


Figure 3: Optimized slices for $S=50,000$, $N=25$