Modeling Temporary Market Impact gt(x) – My Approach

Background

In this task, my objective was to model the **temporary market impact function** gt(x) which represents the average slippage incurred when placing a market order of size 'x'. Slippage, in this context, refers to the difference between the expected execution price (usually the best ask) and the actual volume-weighted average execution price.

The challenge was to use raw Limit Order Book (LOB) data from three tickers—**CRWV**, **SOUN**, and **FROG**—and derive meaningful insights despite limited sample diversity. While not statistically exhaustive, this exercise helped me simulate a real-world scenario of building an execution cost model from scratch.

Step-by-Step Process

1. Parsing and Preprocessing the Data

- I began by scanning and reading CSV files for each ticker. Each file represented one trading day of high-frequency data with depth-10 LOB snapshots.
- For every file:
  - I extracted columns named `ask_px_00`, `ask_sz_00`, ..., `ask_px_09`, representing the top 10 ask levels.
  - I sampled one row every 200 ticks to reduce redundancy and decrease computation time while still capturing representative LOB states.
- For each sampled row (LOB snapshot), I simulated hypothetical buy orders of increasing size, ranging from **10 to 5000 shares**.

2. Simulating Order Execution and Calculating Slippage

For each simulated order size:

- I virtually filled the order by walking up the ask side of the book, level by level, until the entire order was filled or the book was exhausted.
- The cost of execution was calculated as the sum of the fills at each price level.

- The slippage was computed as:
  Slippage= (Total Fill Cost/ Order Size) - Best Ask
- This gave me the temporary impact gt(x) for each snapshot and order size.

By repeating this process across all snapshots and files, I collected hundreds of slippage values per order size and averaged them to obtain a clean and denoised gt(x) curve.

Step 3: Fitting Candidate Models to gt(x)

To analyze how slippage scales with order size, I evaluated three model families:

A. Linear Model:
gt(x)=βx

- This assumes slippage increases proportionally with order size.
- While simple and interpretable, it tends to overestimate slippage at higher sizes and underestimate it at small sizes.

B. Power-Law Model:
gt(x)=ax^b

- Often cited in academic literature.
- Captures *diminishing marginal slippage* (i.e., the rate of slippage increases slows down as order size increases), which aligns with realistic LOB dynamics.

C. Exponential Model:
gt(x)= a (1 - e^{-b x})

- Models slippage as saturating—useful when large order sizes “consume” most of the visible liquidity.

I used SciPy's `curve_fit()` to estimate parameters for each model and calculated **R^2 scores** to compare their fits objectively.

Step 4: Results Across Tickers

CRWV Analysis

When I modeled the temporary market impact for CRWV, the resulting plot told a fairly intuitive story. I observed a **concave shape** in the actual slippage curve — as order size increased, the slippage initially grew rapidly and then plateaued beyond a certain threshold. This suggested some kind of **saturation effect**, where adding more size no longer increased market impact proportionally.

- The **linear fit** clearly underperformed, with an R^2 = -0.197, failing to capture the flattening behavior at larger order sizes.

- The **power-law fit** improved things substantially, achieving an R^2 = 0.846 and better tracked the early steep rise but still missed the eventual plateau.

- The **exponential fit**, with R^2 = 0.983, performed the best. It managed to model the saturation behavior quite effectively. Visually and statistically, this was the most reliable approximation of gt(x) for CRWV.

Conclusion: For CRWV, the **exponential model** most accurately captured the diminishing returns in slippage as order size increased — a plausible behavior in illiquid microcap stocks where initial volume significantly shifts price, but larger blocks don't cause proportionally larger moves.

SOUN Analysis

SOUN was by far the cleanest and most predictable. The slippage curve followed a **smooth, increasing trend** throughout the range of order sizes, without any unexpected dips or saturations.

- The **linear fit** performed decently with an R^2 = 0.967, though it still underestimated curvature.

- The **power-law fit** improved upon this with R^2 = 0.996, closely matching the actual curve across all ranges.

- The **exponential fit** did marginally better with an **almost perfect** R^2 = 0.999. It was able to capture the natural convexity in the relationship and gave me strong confidence in its predictive ability.

Conclusion: For SOUN, both **power-law and exponential models** are valid, but the **exponential model** edges out slightly. This reflects a stable, liquid name where order size has a consistent, nonlinear effect on slippage — ideal conditions for smooth modeling.

FROG Analaysis

FROG's behavior was unexpectedly complex. The slippage curve peaked and then began to **decline** for very large orders. At first glance, this was surprising, as one typically expects a non-decreasing relationship between order size and impact.

- The **linear model** performed the worst here, with a bizarre $R^2 = -0.844$ indicating an almost inverse relationship to actual slippage.

- The **power-law** and **exponential fits** gave R^2 values of **0.227** and **0.465**, respectively — both relatively low.

- More importantly, none of the models were able to capture the **non-monotonic structure** of the actual slippage curve. The drop-off beyond ~2000 shares likely points to an anomaly: possible liquidity injections (e.g., hidden liquidity, midpoint fills, or off-book trades) that reduced the apparent market impact at scale.

Conclusion: For FROG, **none of the standard models** sufficiently described the behavior. I'd consider this an indicator that for certain stocks, modeling gt(x) might require **piecewise fits**, **hybrid nonlinear models**.

Through these experiments, I learned that **modeling temporary market impact is not a one-size-fits-all** process. While exponential and power-law models provide good baselines, real market behavior often involves **hidden microstructure dynamics** that linear or simple parametric models can't capture.

In real trading strategy design, I'd consider adaptive modeling per-ticker — perhaps starting with exponential or power-law fits but validating them against outliers like FROG.

Model Fit of gt(x) Across Tickers