Project

October 31, 2025

1 Project Proposal: Simulating a Limit Order Book (LOB) and Market Dynamics

1.1 Objective

The goal of this project is to simulate the dynamics of a financial market's limit order book (LOB), such as those used in the TSX or NASDAQ, using discrete-event simulation.
The study will model random arrivals of buy and sell orders, cancellations, and executions, and evaluate the performance of different market-making strategies.

Key goals include:

- Understanding how stochastic order arrivals influence market liquidity and price dynamics.
- Measuring execution speed, bid-ask spreads, and volatility.
- Exploring trade-offs between market maker profit, inventory risk, and system efficiency.

1.2 System Description

The system represents an electronic market where traders submit limit and market orders for a single asset.
Each order is modeled as an event in the simulation:

- **Limit orders:** Orders to buy or sell at a specified price; join a price-level queue if not immediately executable.
- **Market orders:** Orders executed immediately at the best available price.
- **Cancellations:** Limit orders may leave the queue before execution (modeled as reneging).
- **Executions:** Orders matched according to price–time priority, removing them from the order book.

The order book is modeled as a set of queues, one for each discrete price level.
Market makers act as servers who post buy/sell orders at different prices.
If a counter-order exists, execution occurs; otherwise, the order waits in the queue.

Key system states include:

- Best bid, best ask, and mid-price
- Queue depth at each price level
- Market maker inventory and profit

The system allows experimentation with:

- Number of market makers (servers)
- Quoting strategies (tight vs. wide spreads, dynamic quotes)
- Arrival rates to simulate volatile periods

1.3 Simulation Parameters

Parameter	Example Setting	Rationale
Buy limit order arrival rate	$\lambda_{\text{buy\_limit}} = 10$ orders/min	Poisson arrivals capture stochastic behavior
Sell limit order arrival rate	$\lambda_{\text{sell\_limit}} = 10$ orders/min	Symmetric market assumption
Market order arrival rate	$\lambda_{\text{market}} = 5$ orders/min	Market orders execute immediately
Cancellation rate	$\mu_{\text{cancel}} = 0.1$/min	Represents reneging or order expiration
Order size	Poisson(1–5) shares	Introduces variability in queue length
Number of market makers	1–3	Server capacity impacts execution speed
Quote placement offset	$\pm k$ ticks from mid-price	Controls liquidity provision and risk exposure

1.4 Methodology

The simulation will be implemented as a **discrete-event system** in Python using `SimPy`, with the following structure:

1. **Initialization:**
   Empty order book, market maker positions, and simulation clock.
2. **Event generation:**
   Random arrivals of buy/sell limit and market orders, plus cancellations.
3. **Event handling:**
   - Market orders execute immediately if matching orders exist.
   - Limit orders join the appropriate price-level queue.

- Cancellations remove orders from queues (reneging).
- Market makers act as servers, posting quotes and executing trades.

4. **Data collection:**
   For each replication, track:
   - Execution time and waiting time distributions
   - Bid-ask spread and mid-price volatility
   - Fill ratio of orders
   - Market maker P&L and inventory levels
5. **Replication:**
   Perform multiple runs with at least five different random seeds to estimate statistics with 95% confidence intervals.

1.5 Analysis

For each scenario (e.g., number of market makers, quoting strategy, arrival rate):

- Compute average and tail metrics across replications.
- Compare performance of different market-making strategies.
- Evaluate trade-offs between execution speed, liquidity, volatility, and profit.
- Examine scenarios under high-order-volume periods to stress-test the system.

1.6 Conclusion

The project will quantitatively evaluate how stochastic order flow and market-making policies affect market performance.

Key expected insights include:

- Faster execution and tighter spreads with more active market makers.
- Trade-offs between market maker profit and risk exposure.
- The impact of arrival rates and cancellations on order execution and liquidity.
- Demonstration of discrete-event simulation as a powerful tool for analyzing financial market microstructure.

Optional recommendations may include: - Optimal server (market maker) allocation,
- Dynamic quoting rules, or
- Priority adjustments to improve execution speed without significantly increasing risk or volatility.

1.7 Future Extensions and Research Directions

If time permits beyond the baseline implementation, we plan to incorporate additional research-oriented extensions to enhance realism and analytical depth:

1. **Adaptive or learning market makers** that adjust spreads dynamically using reinforcement learning or rule-based policies.

2. **Heavy-tailed or self-exciting (Hawkes) order arrivals** to capture volatility clustering and bursty trading behavior.

3. **Latency and information asymmetry effects** to study fairness and instability under heterogeneous agent delays.

4. **Price impact modeling**, where large trades influence the mid-price through nonlinear feedback functions.

5. **Informed versus noise trader dynamics**, linking execution to a latent "fundamental" price process.

6. **Endogenous price formation**, where prices evolve directly from agent interactions and order-book states rather than from an exogenous price process.

The main goal for this course project will be to develop a **simplistic, working limit order book simulation** with correct queueing dynamics, validated through replication and statistical analysis. These extensions represent **potential research enhancements** that we will implement and analyze if time allows.