

Tick Data Generation (Q0)

Assumptions

1. The tick size is fixed at 0.1.
2. The spread between best bid price and best ask price is fixed at 0.1 (1 tick size).
3. Best ask price is simulated using geometric Brownian motion and it's a jump process.
4. Order size is not large enough to impact bid price or ask price. Matching algorithms like Itayose is not simulated in this test.
5. Buy orders will be matched at best ask price, sell orders will be matched at best bid price. Order queue is not simulated in this test. In case of BBO jump, the orders pending in between the last BBO and current BBO are filled at the current BBO accordingly.
6. Trade data is only published when order is filled. And it would then appear in tick data as with information like last price and last quantity.

Formula

For the best ask price, the initial price is donated as S_0 , with S_t represent the ask price at time t . μ affects the drift whereas σ affects the volatility of price movement.

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

$$d\log S_t = \left(\mu - \frac{1}{2}\sigma^2 \right) dt + \sigma dW_t$$

Hence, it satisfies the distribution as following

$$\log S_t \sim N \left(\log S_0, + \left(\mu - \frac{1}{2}\sigma^2 \right) t, \sigma^2 t \right)$$

Parameter Value

To properly simulate the movement of best ask price with geometric Brownian motion, we need to assign proper values to the following parameters: S_0, μ, σ, t .

Parameter	Meaning	Recommend Value
S_0	The very starting best ask price	100
μ	$(\mu - \frac{1}{2}\sigma^2)$ affects the drifting direction and magnitude of price over time. If the value is positive, it means the average price would statistically move in the upward direction over time. Meanwhile, the directional movement should be subtle in magnitude.	0.021
σ	σ^2 affects the volatility of price over time. A larger value means the price movement would be more volatile. And vice visa.	0.2

t	<p>t is the time difference between the current quote and the very beginning one.</p> <p>As the time between any two quotes is independent and generally in the span of several microseconds, we simulate by randomly generating number from 0.01 to 0.05</p>	$t = \sum_{n=1}^N \Delta t$
----------	---	-----------------------------

Order Matching (Q1)

Prior to receive any update from server, the order status would become *"none"* to indicate status is not applicable. Server receives orders from client and compares the order against the current best bid and offer. As we are assuming orders' size are not large enough to impact best bid price or best offer price, we do not compare order size with the best bid size or best ask size.

Orders would be immediately filled if they satisfy either the 2 conditions listed below. In this case, order status would become *"filled"*.

- buy order with price equal to or greater than the best offer price
- sell order with price equal to or smaller than the best bid price

Orders would be kept in order book and waiting to be filled if they satisfy either the 2 conditions listed below. In this case, order status would become *"working"*.

- buy order with price smaller than the best offer price
- sell order with price greater than the best bid price

Market Feed (Q2)

Server would only publish trade data upon filling order with the following information.

Parameter	Details
timestamp	the time when trade is generated
ticker	the instrument ticker
price	the price the order is filled at
size	the size of the order
side	buy sell

Server would publish quote data upon receiving quote from exchange with the following information.

Parameter	Details
timestamp	the time when trade is generated
ticker	the instrument ticker
bid price	the last best bid price
bid size	the last best bid size
ask price	the last best ask price
ask size	the last best ask size
last	the last trade price
turnover	the aggregated trading value in between the previous and current quote

Subscription (Q3 & Q4)

Assuming multiple clients exists, each with an identical account name. They all send their own orders to server while receiving corresponding trades. When subscribing with corresponding account name, we get the specific client's orders / trades data. When subscribing with "server" as account name, we get all clients' orders / trades data.