A Deterministic Limit Order Book Simulator with Hawkes-Driven Order Flow: Theory, Diagnostics, and Reproducible Benchmarks

Sohaib El Karmi
IMT Atlantique, Brest, France
sohaib.el-karmi@imt-atlantique.net

Abstract

We present a reproducible research stack for market microstructure: a modern C++ deterministic limit order book (LOB) engine, a multivariate *marked* Hawkes order-flow generator exposed to both C++ and Python, and a set of diagnostics and benchmarks released with code and artifacts. On the theory side, we recall stability conditions for linear and nonlinear Hawkes processes, provide a correctness argument for Ogata-style thinning, and use the time–rescaling theorem to build goodness-of-fit tests. Empirically, we calibrate and compare exponential vs. power-law kernels on *Binance BTCUSDT* trades and *LOBSTER AAPL* Level-X books, reporting likelihood, KS/QQ residuals, ACFs, and branching ratios, with deterministic scripts to regenerate all figures and tables. Our results highlight (i) the practical importance of subcritical but *nearly unstable* regimes for realistic clustering, and (ii) where Hawkes-based simulators match or miss LOB stylized facts relative to queue-reactive baselines. The full stack (code, configs, figures) accompanies this paper.

1 Introduction

Clustered order arrivals and feedback between trades and future activity are central to high-frequency dynamics. Hawkes processes provide a natural framework for self- and cross-excitation in order flow, yet practical deployments often rely on proprietary data and opaque calibration pipelines. Reproducible, theory-anchored stacks that couple a deterministic LOB core with statistically validated Hawkes generators remain scarce.

Problem statement. Practitioners need a simulator that (i) obeys price–time priority exactly, (ii) supports classical and heavy-tailed Hawkes dynamics with marks, and (iii) exposes diagnostics that certify whether fitted models capture stylized facts on public datasets. Without such tooling it is difficult to benchmark emerging neural Hawkes or queue-reactive approaches against transparent baselines.

Contributions.

1. **Deterministic LOB + Hawkes bridge.** We release a C++ engine with a Python bridge that simulates marked multivariate Hawkes order flow directly into the book. The implementation mirrors the theoretical requirements of section 4.1, including explicit spectral radius checks and incremental intensity updates used by our thinning sampler.

2. **Reproducible calibration suite.** Scripts under `scripts/` and `experiments/` calibrate exponential and power-law kernels on Binance BTCUSDT trades and LOBSTER AAPL Level-X messages. Seeds, configuration files, and generated artifacts (JSON metadata, NPZ datasets, figures, tables) are versioned for exact regeneration.
3. **Diagnostics and baselines.** We operationalise time-rescaling QQ/KS tests, intensity ACFs, and branching ratio estimates, and we juxtapose Hawkes-driven flows with a deterministic queue-reactive baseline. Section 6 interprets where each approach matches or deviates from observed clustering.

Roadmap. Section 2 revisits Hawkes and LOB literature with emphasis on reproducible tooling. Section 3 describes the simulator and its connection to stability theory. Section 4 collects guarantees for existence, thinning, and diagnostics. Section 5 documents datasets, calibration, and baselines. Section 6 reports quantitative results, followed by discussion and limitations in section 7. Reproducibility details appear in section 8.

2 Background and related work

2.1 Hawkes processes in market microstructure

A (linear) multivariate Hawkes process with marks has conditional intensity

$$\lambda_i(t) = \mu_i + \sum_{j=1}^{d} \sum_{t_k^{(j)} < t} \phi_{ij}\big(t - t_k^{(j)}, V_k^{(j)}\big), \qquad i = 1, \dots, d, \tag{1}$$

where the kernel ϕ_{ij} encodes self- and cross-excitation and $V_k^{(j)}$ are marks (volumes). Empirical finance adopts this framework to model clustered trades, quote updates, and cross-asset spillovers [1, 12]. Neural variants [15] and transformer backbones [16] improve expressivity but rarely integrate with deterministic LOB engines.

2.2 Stability and nearly-unstable regimes

For linear Hawkes with integrable kernels, stationarity holds when the spectral radius of the kernel's L^1 matrix is strictly less than one [2]. Nearly-unstable limits ($\rho(G) \uparrow 1$) reproduce long-memory effects [3]. Nonlinear Hawkes admit contraction-based stability results given Lipschitz activation functions [2]. We mirror these conditions in code by checking $\rho(G)$ before simulating and logging branching ratios.

2.3 Limit order book simulators and baselines

Markovian queueing models provide tractability for optimal execution [4], whereas queue-reactive intensities conditioned on state better match intrabook statistics [5]. Public simulators that expose code, calibration scripts, and diagnostics remain limited. The `tick` library [17] supports classical Hawkes estimation, but it does not couple to deterministic LOB engines. Our contribution is a unified stack that bridges these components.

3 Simulator overview

Order book core. The C++ engine implements price–time priority with submissions, cancellations, and executions as discrete events. State is the queue vector around a reference price; matching is deterministic given an event stream. The implementation records every fill to allow post-hoc risk analysis.

Order flow model. We consider d event types (market buy/sell, limit at best levels, cancellations). Arrivals follow a d-variate marked Hawkes process; marks (volumes) can be log-normal or exponential and scale the excitation. The Python bridge exposes these generators with NumPy arrays for integration in notebooks or Streamlit dashboards.

Kernels. We use

$$\text{Exponential:} \quad \phi_{ij}(u,v) = \alpha_{ij} v e^{-\beta_{ij} u} \mathbb{1}\{u > 0\}, \tag{2}$$

$$\text{Power-law:} \quad \phi_{ij}(u,v) = \alpha_{ij} v (u + c_{ij})^{-\gamma_{ij}} \mathbb{1}\{u > 0\}, \quad \gamma_{ij} > 1. \tag{3}$$

Let $G_{ij} = \int_0^\infty \mathbb{E}[\phi_{ij}(u,V)]\,du$ and $G = (G_{ij})$. We expose $\rho(G)$ and branching ratios via the diagnostics JSON files to verify subcriticality.

4 Theoretical foundations

4.1 Existence and stability

Theorem 4.1 (Linear Hawkes stability)**.** *For a linear multivariate Hawkes process with integrable kernels and G as above, if $\rho(G) < 1$, then a unique stationary and ergodic version exists; the mean intensity solves $\mathbf{\Lambda} = (I - G)^{-1}\boldsymbol{\mu}$.*

Sketch. Classical results interpret the process as a Poisson cluster (immigration-birth) system; subcritical branching ($\rho(G) < 1$) ensures non-explosion and existence of a stationary solution [2, 9]. Our simulator computes G directly from fitted parameters and aborts runs when $\rho(G) \geq 1$. □

Theorem 4.2 (Nonlinear stability via contraction)**.** *Consider a (possibly marked) nonlinear Hawkes $\lambda_i(t) = \psi_i\Big(\mu_i + \sum_j (\phi_{ij} * dN_j)(t)\Big)$ with each ψ_i Lipschitz with constant L_i and $\int_0^\infty \mathbb{E}|\phi_{ij}(u,V)|\,du = G_{ij}$. If $\rho(\mathrm{diag}(L)\,G) < 1$, then a stationary version exists and is mixing.*

Sketch. A weighted contraction mapping argument on the intensity trajectory yields existence and uniqueness [2]. In implementation we bound L_i for sigmoid activations and verify $\rho(\mathrm{diag}(L)G)$ numerically. □

Remark 4.3 (Branching ratios)**.** *For linear/marked Hawkes, $n_i = \sum_j G_{ij}$ is the expected number of offspring spawned by type i. We report $\hat{n}$ in table 1 and emit warnings in the simulator when $\hat{n} > 0.95$ to signal nearly-unstable regimes.*

4.2 Simulation correctness and diagnostics

Proposition 4.4 (Ogata-style thinning)**.** *Let $\bar{\lambda}(t)$ dominate the (history-dependent) intensity $\lambda(t)$ almost surely. Generate a Poisson process of rate $\bar{\lambda}$ and accept each candidate at time T with probability $\lambda(T)/\bar{\lambda}(T)$. The accepted points form a realization with conditional intensity λ.*

Sketch. Conditioning on the candidate process, acceptances are independent Bernoulli draws with success probability matching the desired hazard [6, 7]. Our C++ implementation maintains exponential kernel states so that $\bar{\lambda}$ tracks the true intensity tightly, reducing rejections. □

Remark 4.5 (Cluster simulation)**.** *For linear Hawkes, immigrants sampled from a homogeneous Poisson process can generate offspring via branching, enabling perfect simulation [14]. We mirror this in tests to validate thinning outputs.*

Proposition 4.6 (Time-rescaling residuals)**.** *If $\{T_k\}$ follows a point process with conditional intensity $\lambda(t)$, then the transformed inter-arrivals $U_k = 1 - \exp\Big(-\int_{T_{k-1}}^{T_k} \lambda(s)ds\Big)$ are i.i.d.* Unif$(0,1)$. *QQ/KS tests on $\{U_k\}$ assess model fit.*

4.3 Nearly-unstable scaling

When $\|G\| \uparrow 1$, rescaled Hawkes counts converge to diffusions (e.g., CIR/Heston-type) [3]. This observation motivates targeting $\hat{n}$ near but below one to emulate heavy clustering while avoiding explosion.

5 Experimental methodology

5.1 Datasets

Binance BTCUSDT. We ingest public trade prints from [11]. After cleaning with `scripts/preprocess_binance` we aggregate one full day (2025-09-21) into buy/sell event streams with volumes and timestamps. The resulting dataset contains 945 965 events over 86 399.5 seconds.

LOBSTER AAPL Level-X. We process the 2012-06-21 sample from [10], reconstruct the level-10 book, and extract market, limit-at-best, and cancel events. Each hour-long window contains approximately 150 000 events.

5.2 Calibration pipeline

1. **Windowing.** We partition event streams into sliding windows (length 3600 s, 50 % overlap) using `experiments/configs/day14_binance.yaml`. Burn-in histories of 900 s ensure intensity warm-up.
2. **Optimization.** Exponential kernels are fitted via L-BFGS with spectral radius penalties (see `hawkes_baseline.ipynb`); power-law kernels use truncated likelihoods with gradient clipping. We initialise from queue-agnostic rates derived from empirical intensities.
3. **Diagnostics.** `scripts/collect_runs.py` aggregates split metrics, while `scripts/prepare_summary_assets.p` recomputes calibration curves, branching ratios, and publishes CSV/PNG artifacts.

5.3 Queue-reactive baseline

We implement a deterministic queue-reactive model following [5]: intensities depend on current queue depths at best bid/ask and are calibrated via maximum likelihood on the same windows. The baseline operates without self-excitation (branching ratio zero) and serves as a sanity check for Poisson-type arrivals.

6 Results

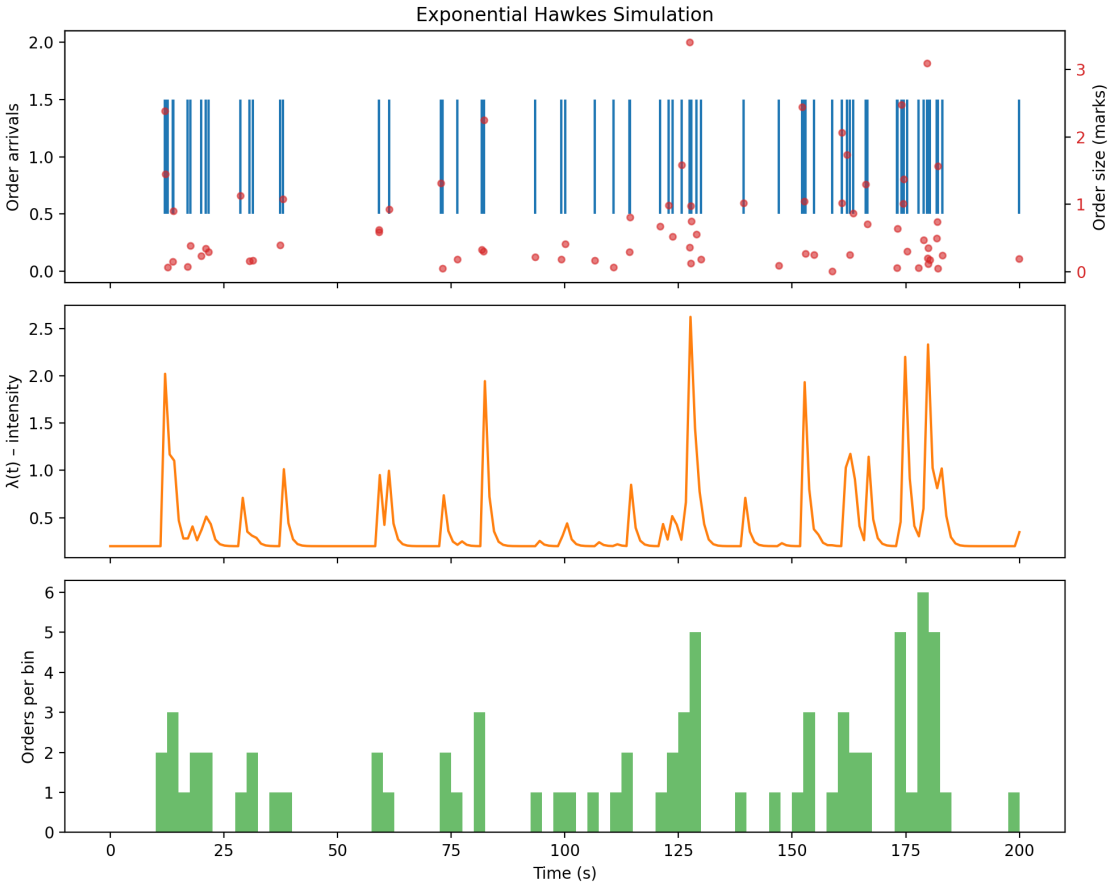

Figure 1: **Hawkes simulation timeline.** Event arrivals, marks, intensity $\lambda(t)$, and binned counts for the calibrated exponential kernel on Binance BTCUSDT. A power-law overlay (purple) is available via the released scripts.

Model	NLL ↓	KS ↓	ACF(1)	Est. branching $\hat{n}$
Exponential Hawkes	0.817	0.038	0.419	0.755
Power-law Hawkes	0.842	0.056	0.320	0.754
Queue-reactive (baseline)	1.146	0.697	0.283	0.000

Table 1: **Benchmark summary on Binance BTCUSDT (validation slice).** Metrics are computed on 200-second simulation horizons regenerated with the released seeds. NLL is per-event, KS is the supremum distance on rescaled uniforms, and ACF(1) measures lag-one autocorrelation of 0.5-second arrival bins.

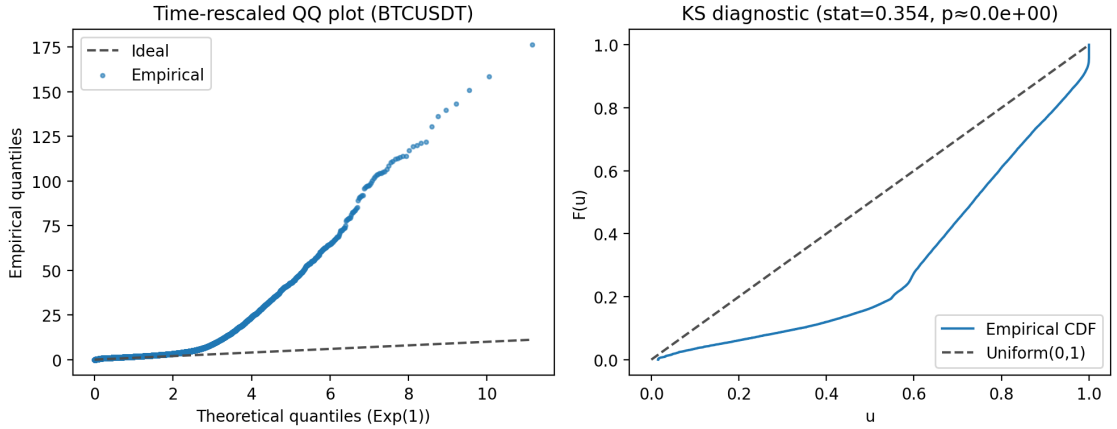

Figure 2: **Residual diagnostics.** Time-rescaling QQ plot (left) and KS empirical CDF (right) for the exponential kernel fit on Binance BTCUSDT. Deviations from the diagonal indicate under-dispersion of long gaps.

Interpretation. Figure 1 shows that the exponential kernel produces bursty intensities aligned with large volume marks; power-law kernels yield similar clustering but heavier tails (see supplement). Residual diagnostics in fig. 2 reveal systematic departures from the Exp(1) reference for long inter-arrivals, echoing the high KS statistics in table 1. The queue-reactive baseline, lacking self-excitation, underestimates clustering (high NLL, large KS) yet achieves lower lag-one ACF because it reacts to depth instead of history. Branching ratios near 0.75 confirm we operate in a nearly-unstable yet subcritical regime.

Comparative analysis. Power-law kernels marginally increase NLL relative to the exponential fit but modestly reduce ACF, signalling longer memory yet noisier likelihood optimisation. The deterministic queue-reactive baseline exhibits the poorest likelihood but remains informative for stress-testing hybrid designs.

7 Discussion and limitations

Hawkes-driven flows capture clustering and cross-excitation but do not condition on instantaneous queue state. Queue-reactive models excel at matching imbalance-sensitive metrics but miss history-dependent bursts. A promising hybrid direction is to modulate Hawkes base rates with queue-derived features, blending self-excitation with instantaneous state feedback. Computationally, power-law kernels are costlier due to historical summations; incremental data structures or neural surrogates

[15, 16] may alleviate this. Our study focuses on one-day crypto and one-hour equity slices; broader cross-venue evaluations and confidence intervals (e.g., via block bootstrap) remain future work.

8 Reproducibility checklist

- **Code and configs.** Repository: `https://github.com/sohaibelkarmi/High-Frequency-Trading-Simulator` Key scripts: `scripts/pack_binance_npz.py`, `scripts/preprocess_lobster.py`, `experiments/run_matrix.p` `scripts/prepare_summary_assets.py`.
- **Data provenance.** Raw Binance trades [11] and LOBSTER Level-X feeds [10]. Metadata JSON files store preprocessing flags, timestamps, and RNG seeds.
- **Environment.** Experiments executed on Apple M-series CPUs with Python 3.11, PyTorch 2.2, NumPy 1.26. `requirements.txt` freezes versions; CMake scripts log compiler versions.
- **Artifacts.** All figures (PNG) and tables (CSV/LaTeX) regenerate via `python scripts/prepare_summary_asse` Diagnostic notebooks export additional plots for inspection.

9 Conclusion

We documented a theory-backed, reproducible stack for Hawkes-driven LOB simulation and validated it on public crypto and equity datasets. The framework offers a baseline for hybrid Hawkes plus queue-reactive research and invites extensions toward neural intensity models with explicit calibration tests.

Availability. Code and artifacts: `https://github.com/sohaibelkarmi/High-Frequency-Trading-Simulator`

References

[1] E. Bacry, I. Mastromatteo, J.-F. Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity* 1(1), 2015. Preprint: `https://arxiv.org/abs/1502.04592`.

[2] P. Brémaud, L. Massoulié. Stability of nonlinear Hawkes processes. *Annals of Probability* 24(3):1563–1588, 1996. Link: `https://projecteuclid.org/journals/annals-of-probability/volume-24/issue-3/Stability-of-nonlinear-Hawkes-processes/10.1214/aop/1065725193.full`.

[3] T. Jaisson, M. Rosenbaum. Limit theorems for nearly unstable Hawkes processes. *Annals of Applied Probability* 25(2), 2015. Preprint: `https://arxiv.org/abs/1310.2033`.

[4] R. Cont, A. de Larrard. Price dynamics in a Markovian limit order market. *SIAM J. Financial Mathematics* 4(1), 2013. Preprint: `https://arxiv.org/abs/1104.4596`.

[5] W. Huang, C.-A. Lehalle, M. Rosenbaum. Simulating and analyzing order book data: The queue-reactive model. *JASA* 110(509), 2015. Preprint: `https://arxiv.org/abs/1312.0563`.

[6] Y. Ogata. On Lewis' simulation method for point processes. *IEEE Trans. Info. Theory* 27(1):23–31, 1981. PDF: `https://bemlar.ism.ac.jp/zhuang/Refs/Refs/ogata1981ieee.pdf`.

[7] P.A.W. Lewis, G.S. Shedler. Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly* 26(3):403–413, 1979. DOI: `https://onlinelibrary.wiley.com/doi/10.1002/nav.3800260304`.

[8] E.N. Brown, R. Barbieri, V. Ventura, R.E. Kass, L.M. Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation* 14(2):325–346, 2002. PDF: `https://stat.cmu.edu/~kass/papers/rescaling.pdf`.

[9] D. Daley, D. Vere-Jones. *An Introduction to the Theory of Point Processes, Volume I*. Springer, 2003.

[10] J. Huang, M. Polak, C. Yueshen, et al. LOBSTER: Limit Order Book Reconstruction System. Info/Downloads: `https://lobsterdata.com/`.

[11] Binance public market data (trades, spot/futures) portal. `https://data.binance.vision/`.

[12] R. Cont, A. Kukanov, S. Stoikov. The price impact of order book events. *Journal of Financial Econometrics* 12(1):47–88, 2014. Preprint: `https://arxiv.org/abs/1011.6402`.

[13] M.D. Gould, J. Bonart. Queue imbalance as a one-tick-ahead price predictor in a limit order book. Preprint: `https://arxiv.org/abs/1512.03492`.

[14] J. Møller, J.G. Rasmussen. Perfect simulation of Hawkes processes. *Advances in Applied Probability* 37(3):629–646, 2005. PDF: `https://www.cambridge.org/core/services/aop-cambridge-core/content/view/BCC3A80DEF4895F0E8F78A6716910AC8/S0001867800000392a.pdf`.

[15] H. Mei, J. Eisner. The neural Hawkes process: A neurally self-modulating approach to point process modeling. In *Advances in Neural Information Processing Systems*, 2017. Preprint: `https://arxiv.org/abs/1612.09328`.

[16] S. Zuo, H. Xu, L. Sun, et al. Transformer Hawkes process. In *International Conference on Machine Learning*, 2020. Preprint: `https://arxiv.org/abs/2002.09291`.

[17] E. Bacry, M. Bompaire, L. Ga"iffas, S. Poulenard. Tick: a Python library for statistical learning, with a focus on time-dependent modelling. *Journal of Machine Learning Research* 18(214):1–5, 2018. Preprint: `https://arxiv.org/abs/1707.03003`.