I. INTRODUCTION

LlamaLend markets are *permissionless* — anyone can deploy them without special approval. However, creating a market requires expertise and responsibility; therefore, this functionality is not included in the default Curve UI and instead uses a factory contract. Before deployment, one should study the behavior of the collateral asset under different market regimes (bullish/bearish trends, volatility spikes) and run simulations. The **LLAMMA simulator** helps to:

- analyze historical price data;
- model different market scenarios under varying parameters;
- assess risk and profitability;
- optimize the key parameters (A, fee, and band count).

Quick Start

It's recommended to use pypy to do faster simulations. Create environment and install dependencies:

```
uv python install pypy@3.11
uv venv --python pypy@3.11
uv sync -p pypy@3.11
```

Add pair to simulator/settings.py and import historical data:

```
python manage.py import_data {pair_name} (i.e. BTCUSDT)
```

II. CONFIGURATION BEFORE SIMULATION

Two main simulations are used for market tuning: `2_simulate_A.py` (search for A) and `1_simulate_dynamic_fee.py` (search for fee). Despite the numbering, you should **first** find A, and **then** find `fee`.

Before running, variables are adjusted for the specific asset (ready-made examples exist in the repository).

Key Variables

- `A=[int(a) for a in logspace(log10(10), log10(300), 50)]` Range of A values scanned in `2_simulate_A.py`: 10 to 300 (50 points). Volatile assets require smaller A; correlated assets (e.g., stablecoins) use larger A.
- `A=30` Fixed A value for `1_simulate_fee.py`.
- `range_size=4` Number of bands. A minimum of 4 is recommended — representing the most conservative stress case.
- `dynamic_fee=0.002` Fee for the initial A scan, calculated by difference of $(p_{oracle} - p) * d\_fee$.
- `fee=logspace(log10(0.0001), log10(0.02), 20)` Range of fees for scanning during `1_simulate_fee.py`.
- `min_loan_duration, max_loan_duration` Time in days. Example: $1/24/2 = 1/48$ days ≈ 30 minutes. Longer duration = slower arbitrage.
- `samples=500000, n_top_samples=5` Number of Monte Carlo samples and averaging window (tail risk). For fee scans, you can lower `samples` and increase `n_top_samples` (e.g. 10^4 and 10^3).
- `add_reverse=true` Adds mirrored price data (balances up/down trends).
- `Texp=600` EMA horizon (seconds), must match the oracle parameter (default NG pools = 600).

III. SIMULATION AND PARAMETER OPTIMIZATION

A. Optimal Amplification Parameter (A)

Run:

```
python3 simulator/pairs/btcusd/simulate_a.py
```

The script plots two charts: losses and discount (for a given band count). The optimal A is at the minimum of the orange curve (typically around $A \approx 30$).

B. Optimal Fee

Run:

```
python3 simulator/pairs/btcusd/simulate_dynamic_fee.py
```

The optimal fee corresponds to the minimum of the loss curve at fixed A and band count.

C. Liquidation and Loan Discounts

The `liquidation_discount` is taken as the y-value of the blue line under the orange curve's minimum (e.g. ≈ 0.07). Then:

$$\texttt{liquidation\_discount} \approx 0.07$$
$$\texttt{loan\_discount} \approx \texttt{liquidation\_discount} + 0.03 \approx 0.10$$