# Liquidity Module Economic Simulation

## Abstract

Liquidity Module has unique methodologies compared to the broadly used CPMM (Constant Product Market Maker) AMM(Automated Market Maker), which is represented by the "Uniswap Model". In this report, we conduct economic simulations to analyze the economic impact of two unique characteristics of the Liquidity Module:

- Batch Execution and
- ESPM (Equivalent Swap Price Model).

Our simulation results suggest that in most market situations, Liquidity Module pool investors enjoy significantly larger returns compared to the CPMM model.

## 1. Objectives

The aim of this simulation is an analysis, using some high-level assumptions, of the economic model used by the Liquidity Module. This model is called the Equivalent Swap Price Model (ESPM) and is described in this associated Litepaper.

Aims of the simulation include:

- Analysis of the economic returns under different parameters and assumptions.
- Benchmarking the performance of the model against the CPMM model (Constant Product Market Maker).
- Assess the sensitivity of pool investor returns to changes in model parameters.
- Identify economic shortfalls in ESPM and propose possible solutions.

This analysis does not aim to be an exhaustive model assessment, but merely for indicative purposes, with a view to encouraging further analysis.

## 2. Methodology

This analysis was performed using an "agent" based approach with the following actors:

- An AMM, which can use either a CPMM approach or the ESPM with a pool price solely decided by such orders submitted by each participant type.
- An external random process called the "Global Price" which represents an observed independent price for a token pair. These follow a 1-day period and simulated 100 times for each scenario. The Price also includes jumps to represent sudden changes in volatility.
- Ordinary traders who submit orders in order to achieve some economic goal such as an investment, hedging, borrowing etc.
- Arbitrageurs whose aim is to make a profit through trading differences between the

quoted pool price and the Global Price

Each of the different actors is modelled using a set of parameters. These are adjusted in turn to perform the analysis. In total three sets of tests are performed:

- Adjusting the Global price process by
    - Changing the volatility of the process
    - Varying the number and size of the jumps
- Changing fee rates and assessing returns under different trading volume scenarios
- Modelling different levels of arbitrage

For each test, the Global price is simulated and the returns collated. Each test is repeated in turn using the CPMM and ESPM AMM's. In the end, the returns are compared and analyzed.

In order to make the test as easy to perform, calculations are done using a Python implementation, which can be found here.

## 3. Model Comparison

This investigation compares two models: ESPM and CPMM. These are defined thus:

- Default Model : ESPM(Equivalent Swap Price Model) + Batch Execution
    - swapPrice = (X + 2*dX) / (Y + 2*dY)
    - Liquidity Module utilizes batch execution which results in price competition among arbitrageurs. the arbCompetitionGauge parameter determines the competition level.
- Comparison Target : CPMM(Constant Product Market Maker) + Sequential Execution
    - swapPrice = (X + dX) / (Y + dY)
    - no price competition among arbitrageurs

## 4. Assumptions

In order to perform the simulation, a number of assumptions need to be made.

The main assumption is that the Global price is randomly generated with fixed volatility and price jumps. In particular, it is assumed that it follows a random walk and that the return of the global price follows a lognormal distribution with a given volatility parameter. It is also assumed that the price also experiences jumps in price and these are fixed in size and occur at a predefined frequency.

It is also assumed that ordinary traders are submitting orders randomly, but around the global price. The total amount of orders are defined using the parameters defined below.

Finally, Arbitrageurs are assumed to submit orders solely to maximize their trading profits. These arbitrageurs compete with each other for this profit. These behaviours are controlled through a parameter-based approach. These parameters are defined below.

## 5. Simulation Parameters

The values for each test are defined here. In most cases each parameter has two values listed: the default value indicates the value used in the control case. Simulation values indicate the range of values used for the given test.

### Fixed parameters

These parameters are the same for all tests.

- numberOfSimulation: number of global price path generation scenarios
  - default: 100
- simBlockSize: all states and environments are updated with this size of blocks (the unit size of each step of simulations)
  - default: 1
- simSeconds: length of a simulation
  - default: 24×*60*×60 (1 day)
- randomOrderSize: the average of order amount from ordinary traders represented by the ratio of the order amount and the pool size
  - default: 0.0001

### Test One: Global Price Volatility Scenarios

- vol: annualized volatility for the global price
  - default: 1.5
  - simulation: 1 / 1.5 / 2
- priceJumpPerDay: number of price jump per day
  - default: 10
  - simulation: 5 / 10 / 20
- priceJumpMagnitude(%): the magnitude of the price jump
  - default: 0.02
  - simulation: 0.01 / 0.02 / 0.03

### Test Two: Pool Return Sensitivity

- feeRate (%): swap fee rate
  - default: 0.003
  - simulation: 0.002 / 0.003 / 0.004
- tradingVolumePerDay (%): daily trading volume assumption from ordinary traders represented by the ratio of trading volume and the pool size
  - default: 0.2
  - simulation: 0.1 / 0.2 / 0.3

### Test Three: Arbitrageur scenarios

- arbTrigger (%): arbitrageurs submit arbitrage orders when the difference between pool price and the global price is larger than arbTrigger (%)
  - default: 0.015
  - simulation: 0.01 / 0.015 / 0.02

- arbCompetitionGauge: competition among arbitrageurs represented by 0~1. higher competition gauge results in swap price nearer to global price.
    - default: 0.5
    - simulation: 0.25 / 0.5 / 0.75

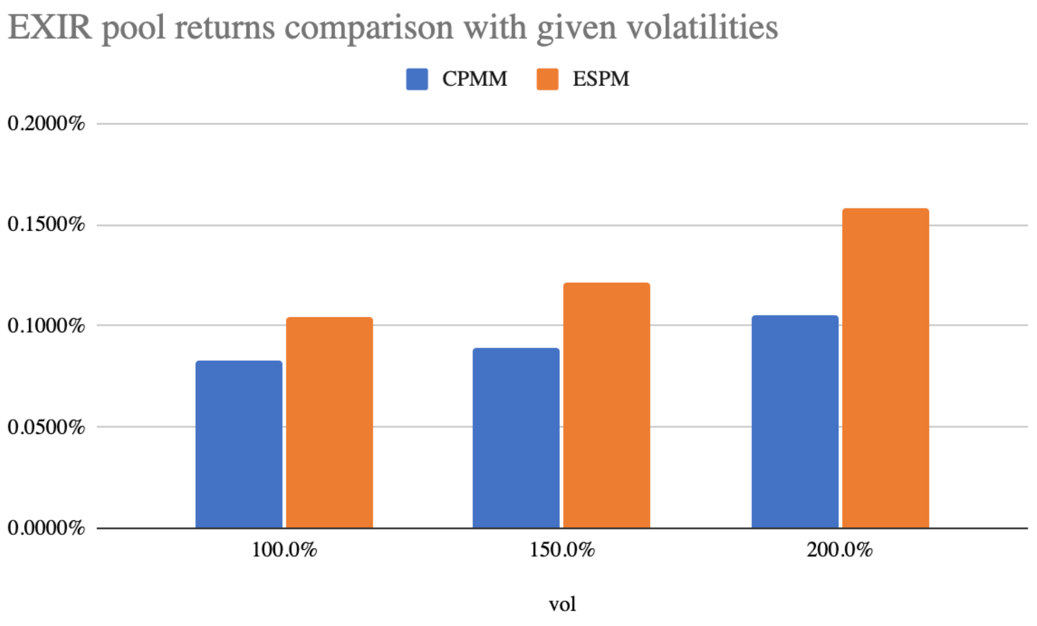
## 6. Simulation Results

### Performance Measure Definitions

In order to perform the comparison, we define the following measures:

- Pool Return: return rate of the pool in a given scenario
- Impermanent Return: a part of pool return rate "expected" from a given price change
- Pool Return EX IR: Pool Return - Impermanent Return
    - The additional return of the pool beside the expected impermanent return
- Arb Trading Volume: total trading volume execute from arbitrageurs / initial pool size
- Arb Return: the total return of arbitrageurs / initial pool size

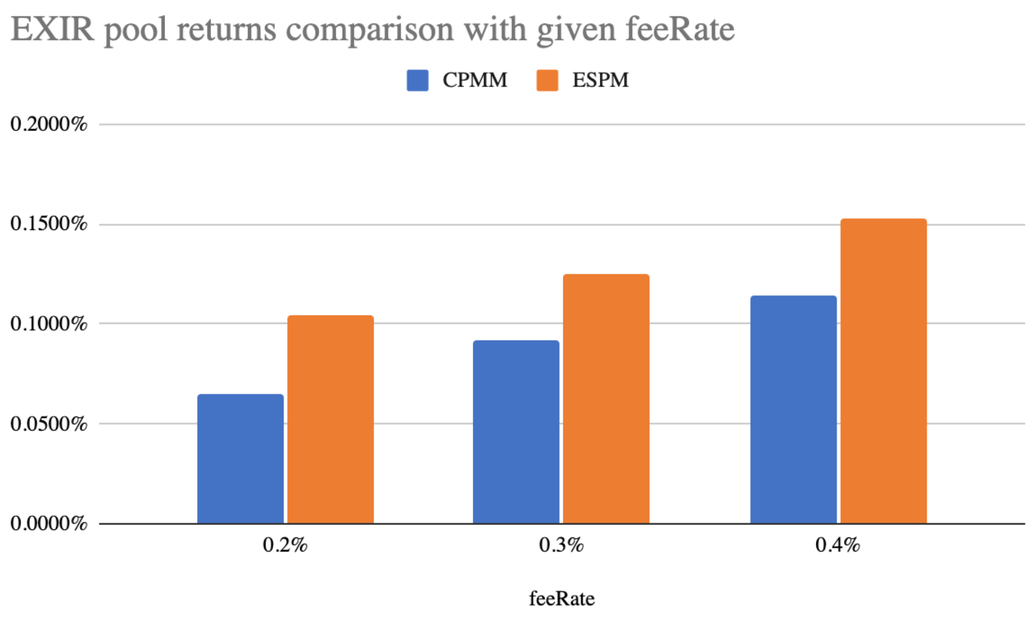
### Test One: Global Price Volatility Scenarios



- High volatilities result in higher EXIR returns for the pool.
- Compensation of higher volatility (200% vs 100%) is 2.34 (0.0532% vs 0.0227%) time greater in ESPM than in CPMM, but the pool return is not fully compensated for the increased impermanent loss risk in higher volatility case.
- Pool investors who deposit tokens to pools with higher volatility reserve tokens need additional incentives to compensate for much higher impermanent loss risk.

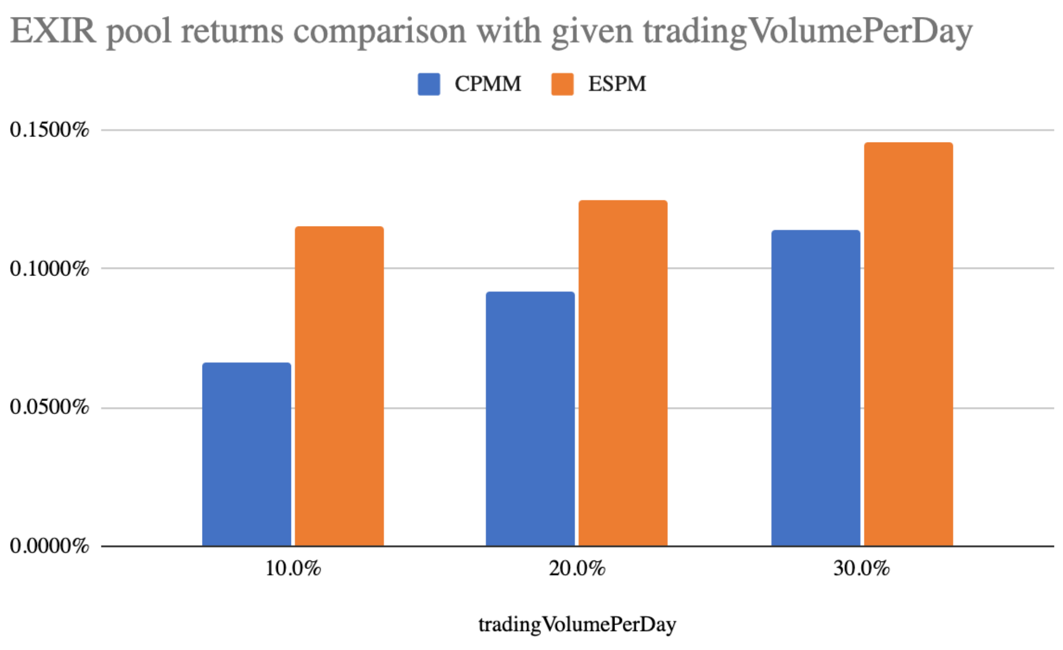
## Test Two: Pool Return Sensitivity

### Sensitivity of returns to changes in fee rates



In this case, there is a linear relationship between feeRate and EXIR pool returns under the CPMM model. However, under ESPM, the effect of higher feeRate is less than CPMM's because its source of return is not only the fees but includes slippage costs accumulated from traders

### Sensitivity of returns to changes in trading volumes

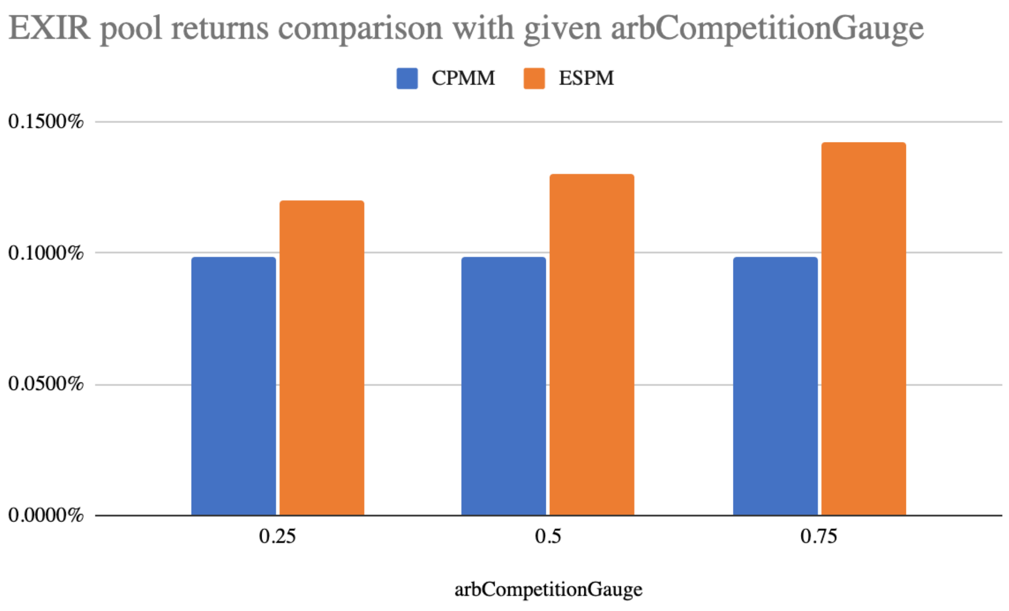


Larger daily trading volume results in larger EXIR pool returns, but the increment is less than the naive expectation of increased fee returns. It is because increased trading volumes due to ordinary traders results in more orders around the global price, which reduces arbitrage trading opportunities and volumes. It also results in significantly less arbitrage volume because ordinary traders tend to trade by taking advantage of the price difference of pool price and global price, hence less arbitrage opportunities.

### Test Three: Arbitrageur scenarios

- arbTrigger - more tight and frequent arbitrage activities are good for both arbitrageurs and pool investors

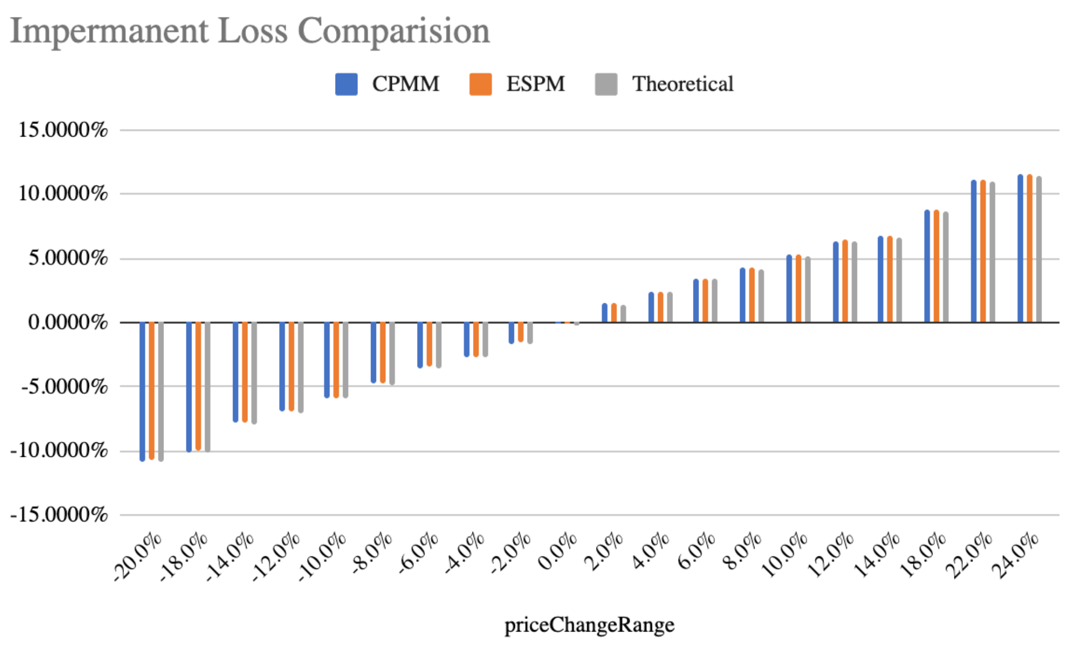
Effect of changes in arbitrageur competition



Here, tighter competition among arbitrageurs result in significantly less returns for arbitrageurs, which result in higher EXIR pool returns for ESPM

### Impermanent Loss Comparison

This graph compares how both models perform in terms of impermanent loss compared to the theoretical calculation:



Our simulation results show that both models have a very accurate amount of impermanent loss compared to the theoretical expectation

## 7. Conclusion

Based on the above analysis, it can be concluded that

- **ESPM outperforms the CPMM model in all the analysed scenarios**, especially in situations where there high volatility, low fee scenarios or low trading volumes. On average, daily 0.0264%, annually 9.64%
- Unlike CPMM, ESPM model returns are **less sensitive to fee structure** because there are other multiple sources of income from realized volatility. Hence, the increasing fee is less efficient for ESPM to gain more profit from pool investors
- **ESPM yields increased pool returns when there are high levels of competition between arbitrageurs**. This is not possible in CPMM.
- **ESPM can provide better returns in high volatility scenarios** since higher volatility results in more active arbitrage trading and ESPM is better at accumulating the higher slippage costs that result from such trading when compared to CPMM. Therefore, ESPM provides a good alternative return source when the volatility of the price increases.

However, it is also noted that pool investors do not earn higher returns do in cases where there is higher volatility. This is because pools with higher volatility tokens have significantly higher impermanent loss risk, but it is offset by higher returns. This is particularly an issue for new tokens as these tend to have relatively higher volatility, but in order to launch successfully, they need a liquid marketplace for token holders to trade.

To overcome this and promote new token liquidity pools, an additional incentive mechanism is required. A possible solution is the use of weighted pools to reduce impermanent risk. A Weighted Pool is a type of liquidity pool that defines a weight for each reserve token so that pool investors can deposit tokens in the pool with a larger USD value of one reserve token, and a smaller USD value of the other reserve token.