



MA/BA/SA:

## Simulating Liquidity Allocation in Decentralized Exchanges

The usage of blockchain-based applications running on Ethereum and other blockchains has skyrocketed in the past year. This is in large part due to the rise of Decentralized Finance (DeFi) applications.<sup>1</sup> These allow users to perform a number of financial activities, e.g. trading and lending, in a decentralized manner. Among the most popular DeFi applications are Decentralized Exchanges (DEXs) such as Uniswap, SushiSwap or Curve. The centerpieces of these DEXs are liquidity pools which are smart contracts running on a blockchain. On the one hand, traders can swap one kind of token for another with these liquidity pools. On the other hand, liquidity providers can deposit their capital into liquidity pools and earn a share of the trading fees.

DEXs have recently seen explosive growth in about any measurable metric and, by now, the number of liquidity pools is almost countless. But how do all these exchanges and pools compete with each other for trading volume and liquidity? How will the sizes and fees of these pools develop in the long run? Can we predict how liquidity will move between the pools and possibly reach an equilibrium at some point? In this project, we will simulate the fee setting and liquidity allocation in



pools using a number of models making different assumptions. We will do so by using past pool and trading data and be building on previous work<sup>2</sup>.

**Requirements:** This project will involve programming in a language of your choice, preferably Python. Expect to be working with a fair amount of math. Interest in finance and blockchain is a plus. We will have weekly meetings to discuss open questions and determine the next steps.

**Interested? Please contact us for more details!**

### Contacts

- Robin Fritsch: [rfritsch@ethz.ch](mailto:rfritsch@ethz.ch), ETZ G94

---

<sup>1</sup>For an introduction into DeFi check out this course: <https://defi-learning.org/>

<sup>2</sup><https://arxiv.org/abs/2105.13510>