Lecture 3: Decentralized Exchanges

Guillermo Angeris

June 2022

Outline

Recap and overview

Interfaces and examples

The trading function

The arbitrage problem

Previous lecture

- Covered some interesting ‘basic’ applications
- And how they can go wrong
- Basic (but popular) notions including NFTs, voting, *etc.*

Lessons(?)

- Governance is very useful!
- But usually not a substitute for solid mechanism design
- Many examples of good governance
- (we will see some in a later lecture)

This lecture

- Cover constant function market makers
- A basic building block of DeFi
- Also one of the largest applications in DeFi!
- Will start being fairly mathy (finally...)

Quick note

- Please fill out the survey if you haven't yet!

Outline

Recap and overview

Interfaces and examples

The trading function

The arbitrage problem

Decentralized exchange interface

- As usual, specify the contract interface:

```
addLiquidity(maxAmounts: vec[uint], env)

removeLiquidity(minAmounts: vec[uint], env)

swap(amountIn: vec[uint], amountOut: vec[uint])
```

- `uint` is shorthand for `unsigned int` (nonnegative number)
- `vec[type]` is a *vector* with elements of type `type`

Decentralized exchange interface (cont.)

- We will focus on the `swap` method today

```
swap(amountIn: vec[uint], amountOut: vec[uint])
```

- In this case `amountIn` and `amountOut` are n-vectors
- `amountIn`: how much of each token you put in
- `amountOut`: how much of each token you expect

Other notes

- We will assume that this 'market' already is initialized
- Has some amounts of liquidity
- We will not (yet) discuss why people would do this
- Or how...

Many implementations

- Simple possibilities
  - `swap` accepts any trade (bad!)
  - `swap` accepts trades at a fixed exchange rate
- Traditionally, `swap` implements an *order book*

Decentralized exchanges

- In many cases *decentralized exchanges* (DEXs) are implemented in a specific way
- Let `amountIn` be $\Delta \in \mathbf{R}_+^n$, and `amountOut` be $\Lambda \in \mathbf{R}_+^n$
- Define some function $\varphi : \mathbf{R}_+^n \to \mathbf{R}$ called the *trading function*
- `swap` accepts if

$$\varphi(R + \gamma\Delta - \Lambda) \geq \varphi(R)$$

  where $0 < \gamma \leq 1$ is a *trading fee* and $R \in \mathbf{R}_+^n$ are the *reserves*
- Reserves are updated as: $R \leftarrow R + \Delta - \Lambda$, if accepted

Constant function market makers

- Any DEX implemented this way is called a *constant function market maker* or CFMM

Decentralized exchanges (cont.)

- We will talk later about how reserves R are related to adding/removing liquidity
- Many of the things we will show carry over for the more general case

$$\psi(R, \Delta, \Lambda) \geq 0$$

- But almost all DEXs use the 'simpler' form

Decentralized exchanges (cont.)

- We will talk later about how reserves R are related to adding/removing liquidity
- Many of the things we will show carry over for the more general case
$$\psi(R, \Delta, \Lambda) \geq 0$$
- But almost all DEXs use the 'simpler' form
- This lecture (mostly) focuses on the 'fee-less' case $\gamma = 1$
- (See homework for more general cases!)

Outline

Recap and overview

Interfaces and examples

The trading function

The arbitrage problem

The trading function

- We will assume two things that hold in practice:
  (a) φ is concave
  (b) φ is nondecreasing
- Concavity lets us say something about ease of solving certain problems
- Nondecreasing will let us say more (see homework!)

Examples

- Many trading functions in practice
- $n = 2$ case is the most common!
- Examples:
  - Sum $\varphi(R) = \mathbf{1}^T R$
  - Product $\varphi(R) = (R_1 R_2 \dots R_n)^{1/n}$
  - Geometric mean $\varphi(R) = R_1^{w_1} R_2^{w_2} \dots R_n^{w_n}$ with $w \geq 0$, $\mathbf{1}^T w = 1$
  - Many, many more...

Some simple consequences

- Any *trade* (Δ, Λ) never decreases $\varphi(R)$
- 'Reasonable' actors will always have equality:

  $$\varphi(R + \gamma\Delta - \Lambda) = \varphi(R),$$

  (see homework for exact statement)
- Approximate exchange rate is $P = \nabla\varphi(R)$ and

  $$P^T\Lambda \leq \gamma P^T\Delta$$

Some simple consequences (cont.)

- Nonoverlapping trades implies

$$\Lambda \leq R$$

- Support is nonoverlapping for 'reasonable' trades

$$\Delta_i \Lambda_i = 0, \quad i = 1, \ldots, n$$

- Trader 'loses' at least

$$(1 - \gamma) P^T \Delta$$

amount per trade

Outline

Recap and overview

Interfaces and examples

The trading function

The arbitrage problem

Arbitrage set up

- Game between market and arbitrageur
- Given an external 'reference' market with prices $c \in \mathbf{R}^n_{++}$
- What is the maximum profit that an agent can make?

Arbitrage problem

- The *arbitrage problem*:

$$\begin{array}{ll} \text{maximize} & c^T(\Lambda - \Delta) \\ \text{subject to} & \varphi(R + \gamma\Delta - \Lambda) \geq \varphi(R) \\ & \Delta, \Lambda \geq 0 \end{array}$$

with variables $\Delta, \Lambda \in \mathbf{R}^n$.

Arbitrage problem properties

- Convex problem ($\approx$ easy) when φ is concave
- Has a simple rewriting in the case with $\gamma = 1$:

$$\begin{array}{ll} \text{minimize} & c^T R' \\ \text{subject to} & \varphi(R') \geq \varphi(R) \end{array}$$

with variable $R' \in \mathbf{R}^n$

- (Arbitrageurs minimize the value of reserves in the contract!)

Properties of solutions

- Using first order optimality conditions:

$$c = \lambda \nabla \varphi(R')$$

  for $\lambda \geq 0$

- *i.e.*, 'marginal price' will match external market!
- See homework for case with fees

Properties of solutions (cont.)

- Let $V(c)$ be the optimal value of the no-fee arbitrage problem
- V has the interpretation of 'portfolio value' after arbitrage
- Useful way of looking at behavior
- In general, every trading function φ has V which is
  - (1-)Homogeneous
  - Nondecreasing
  - Concave
- Is the opposite also true...?

Next lecture

- We will take a look at more 'sophisticated' results
- Also see where two-coin case is interesting
- Construct some interesting payoffs and how liquidity provision works
- Building blocks for MEV, many-market arb, among others