Elements of DeFi

https://web3.princeton.edu/elements-of-defi/

Professor Pramod Viswanath

Princeton University

Lecture 7:
Improving CFMMs

Last Lecture: CFMMs and their properties

- Look at CFMMs from trader's perspective
  - Pricing
  - Slippage
  - Arbitrage
  - Relation with curvature

- Look at CFMMs from liquidity provider's perspective
  - Impermanent Loss
  - Arbitrage Loss
  - Picking the bonding curve
  - Fees

This lecture: Improving CFMMs

- Make CFMMs more capital efficient
  - LP's POV : Concentrated liquidity – move liquidity around
  - Trader's POV : DEX aggregators – Batching + Routing – avoid arbitrage losses
- Private CFMMs – to avoid MEV
- CFMMs as derivatives

Recall : CFMMs

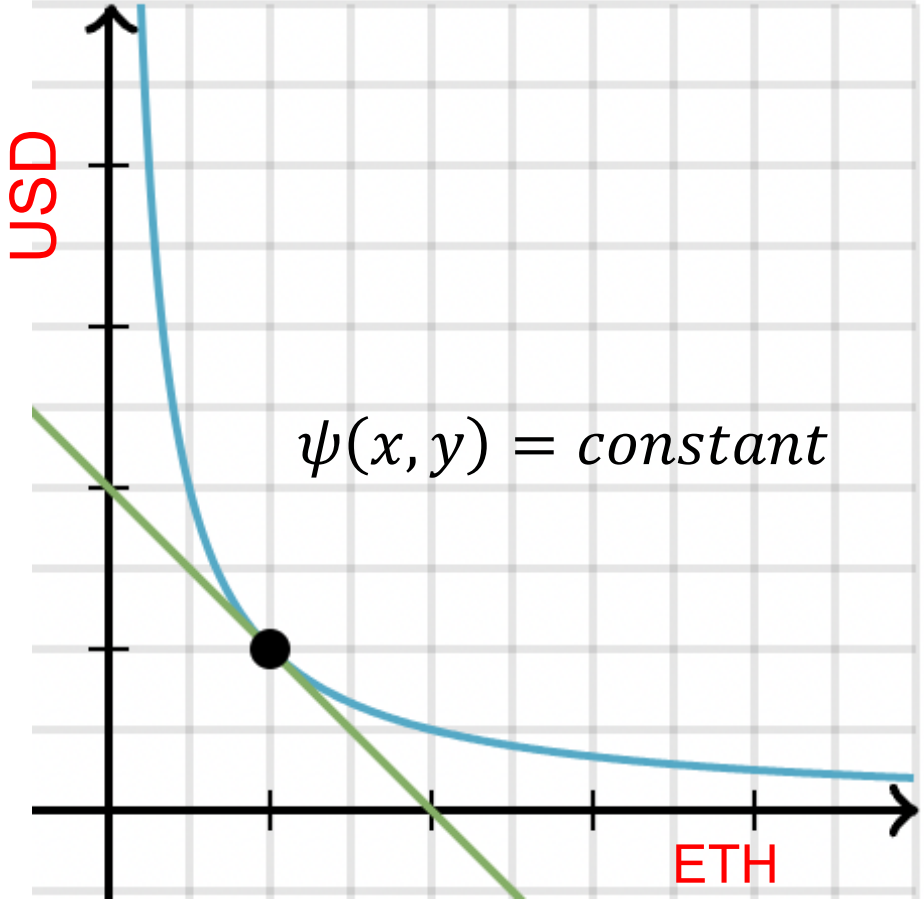
- CFMM: Constant Function Market Makers
- Use Bonding Curves to constrain reserves

$$\psi(x, y) = \psi(x + \Delta_x, y - \Delta_y)$$

OR

$$\psi(x, y) = constant$$

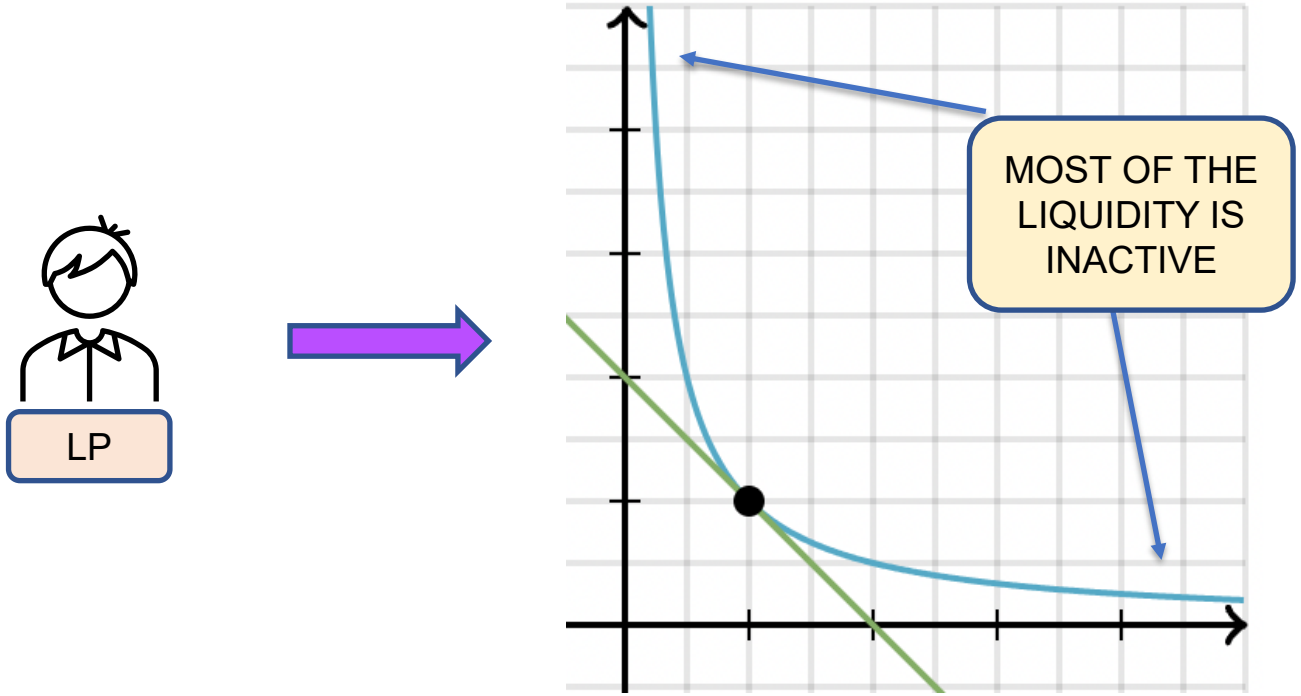
- Slope of the tangent = Price



Recall: Problems - Capital Inefficiency

Capital inefficiency : Less capital efficiency than LOBs - why?

- LPs cannot move liquidity around
- Was possible in LOBs



Capital Inefficiency leads to Arbitrage Loss

Arbitrage loss :

- *increases* with volatility (recall in the tradfi case, more volatility gave more profit) -> Fees have to give a return *and* cover these losses

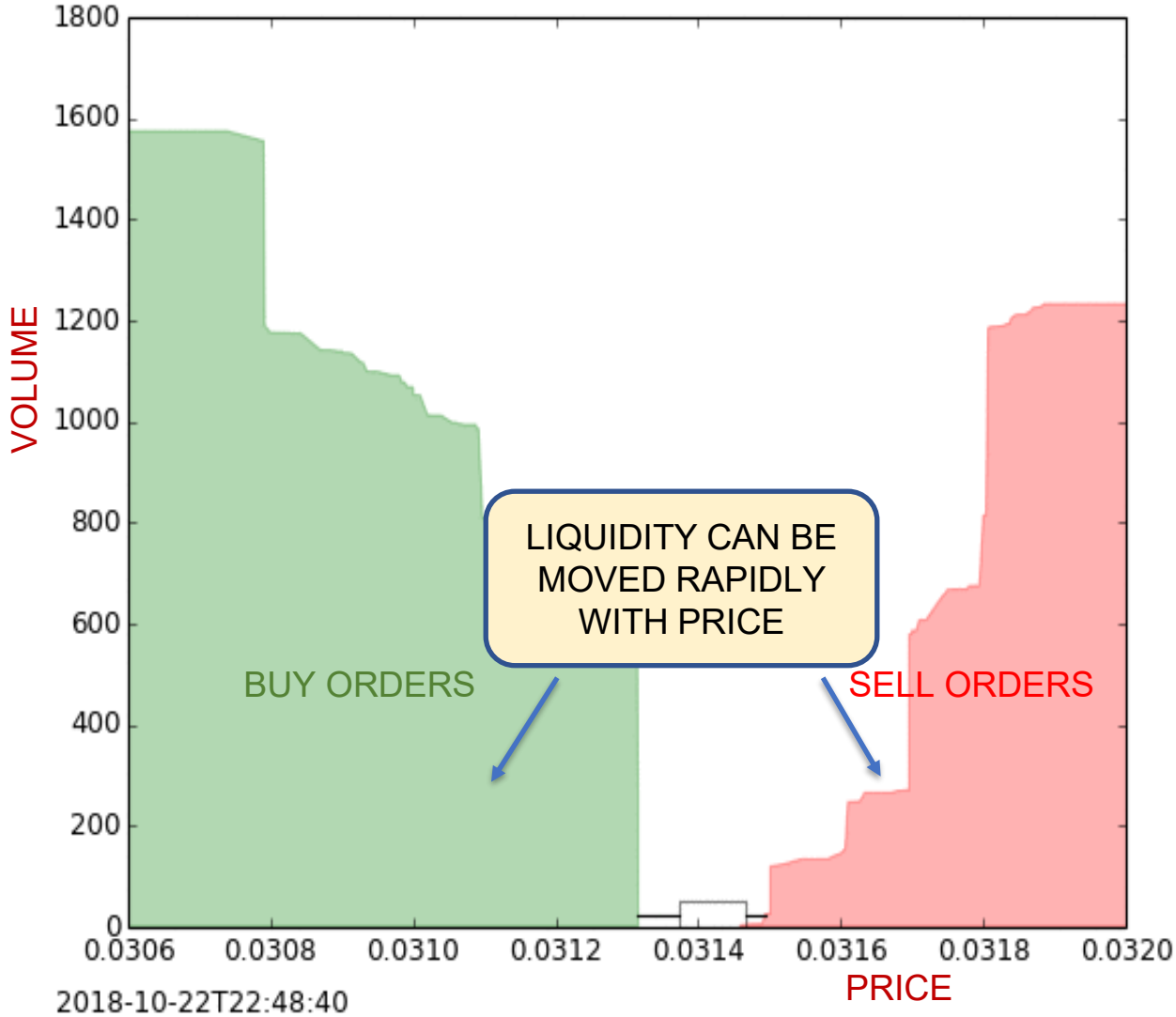
- LPs are sitting ducks – easily fleeced by arbitrageurs

- “If I see a Uniswap LP in the wild, I go up to them, shake their hand and thank them for their service”

  - Mark Twain (probably)

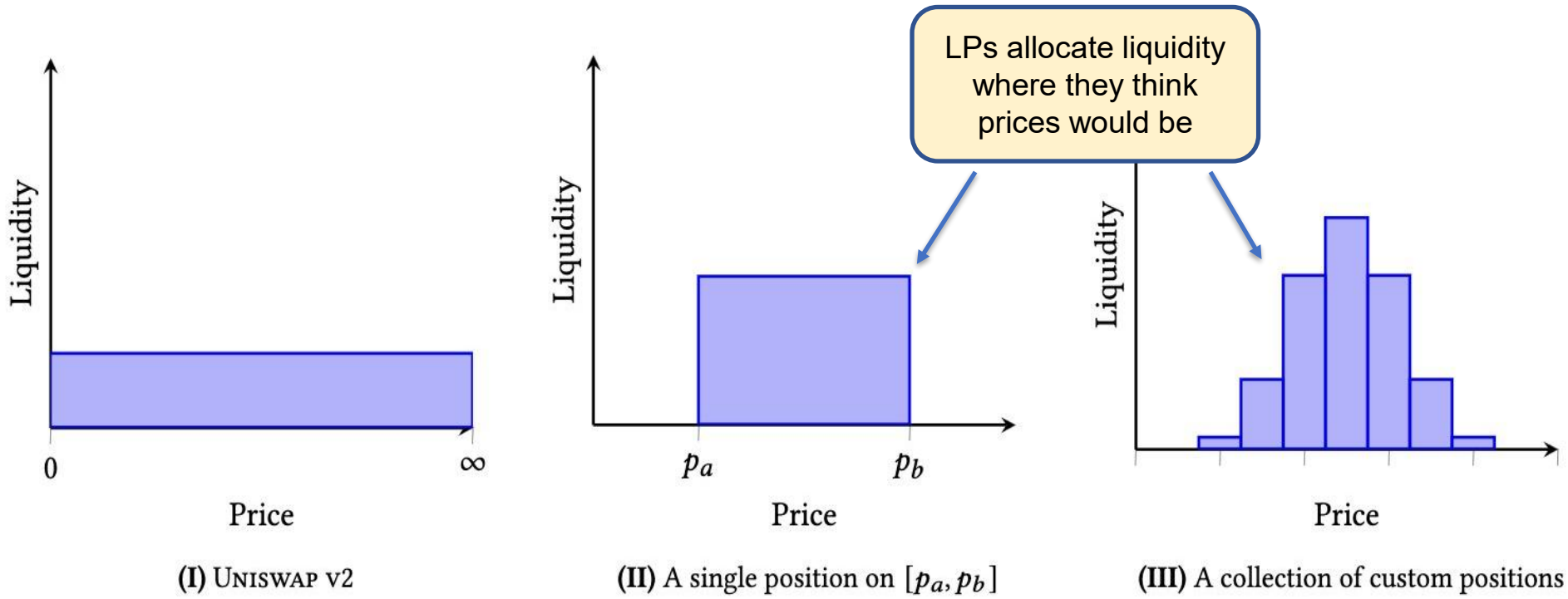
Capital Efficiency in LOBs

- How is capital efficiency achieved in LOBs?
- LPs can move/cancel orders around without paying fees
- Not possible in a fixed bonding curve CFMMs
- Same liquidity serves all prices



Solution: Concentrated Liquidity

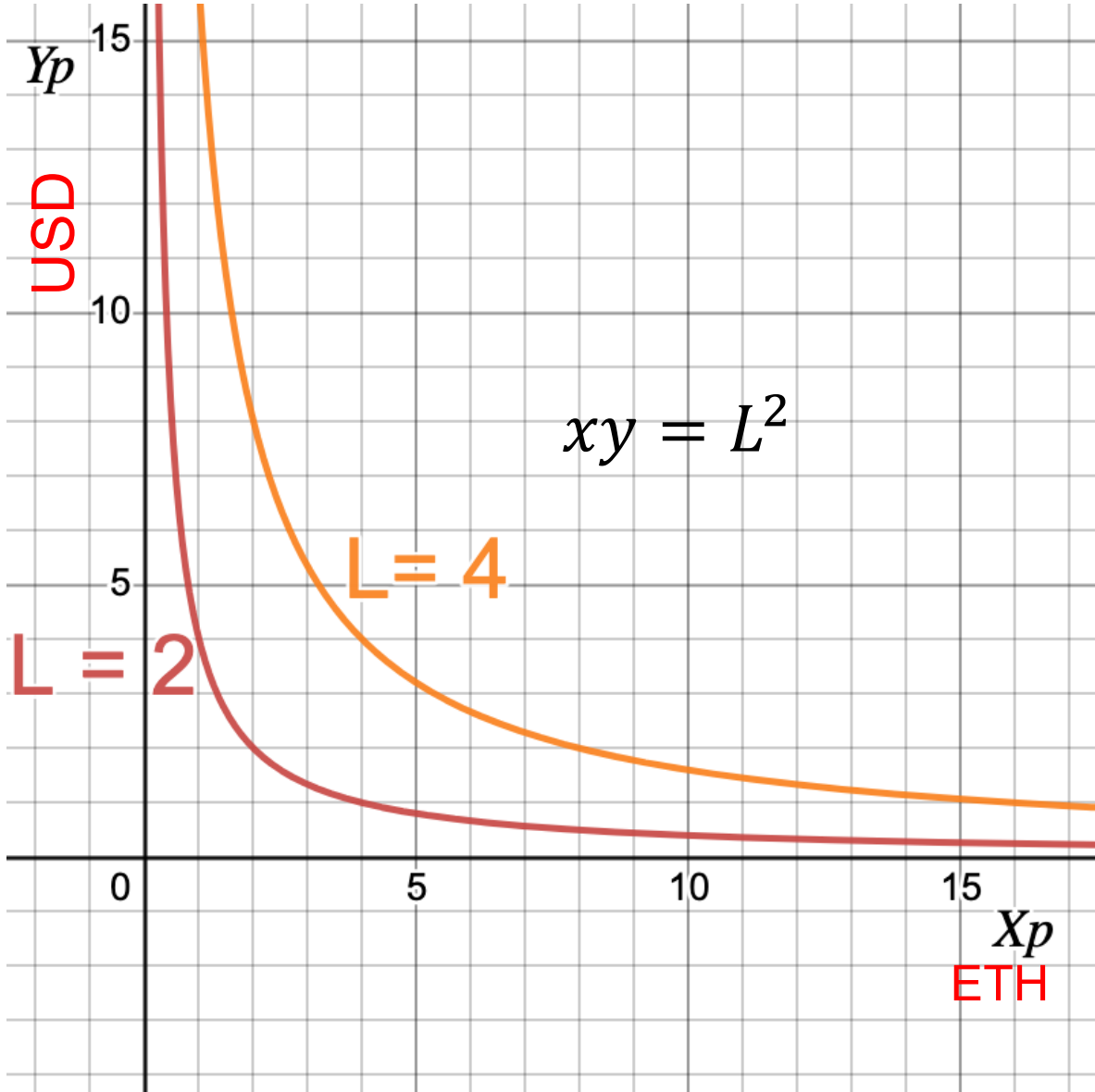
- Allow LPs to specify range of prices
- Divide price range into “buckets”
- LPs choose how much liquidity to allocate to which bucket



(I) UNISWAP V2 **(II)** A single position on $[p_a, p_b]$ **(III)** A collection of custom positions

Measuring liquidity

- Before we look at how to enable LPs to distribute liquidity, need a good measure
- Constant Product MMs – has many favorable properties
- L is a good measure of liquidity – indicates depth of the market



Measuring liquidity

- Also, L is additive – why?
- We know that :

$$xy = L^2$$

$$p = \frac{y}{x}$$

- Express x, y in terms of p,L

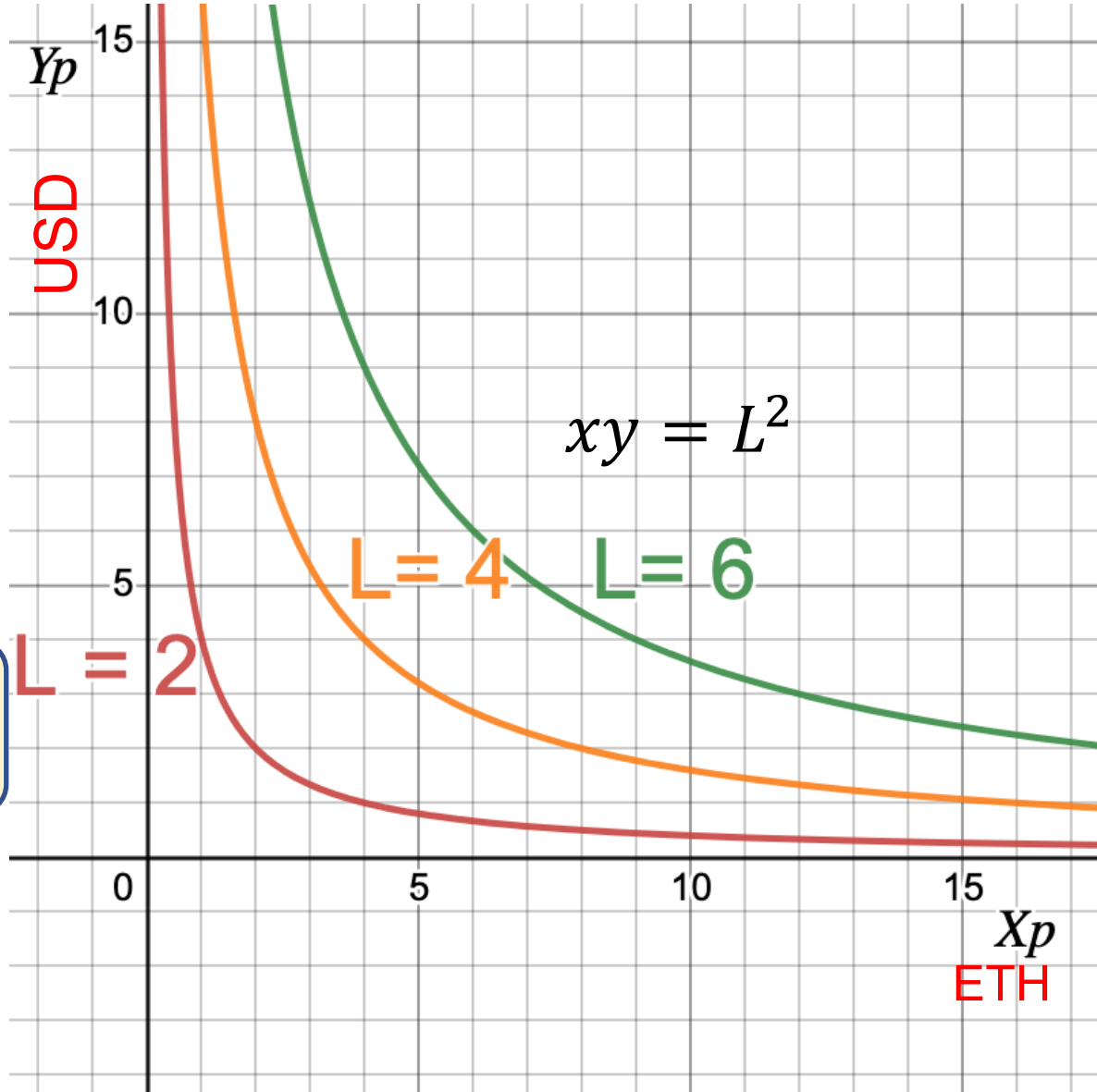
$$x = \frac{L}{\sqrt{p}}$$

$$y = L\sqrt{p}$$

Reserves are linear in L

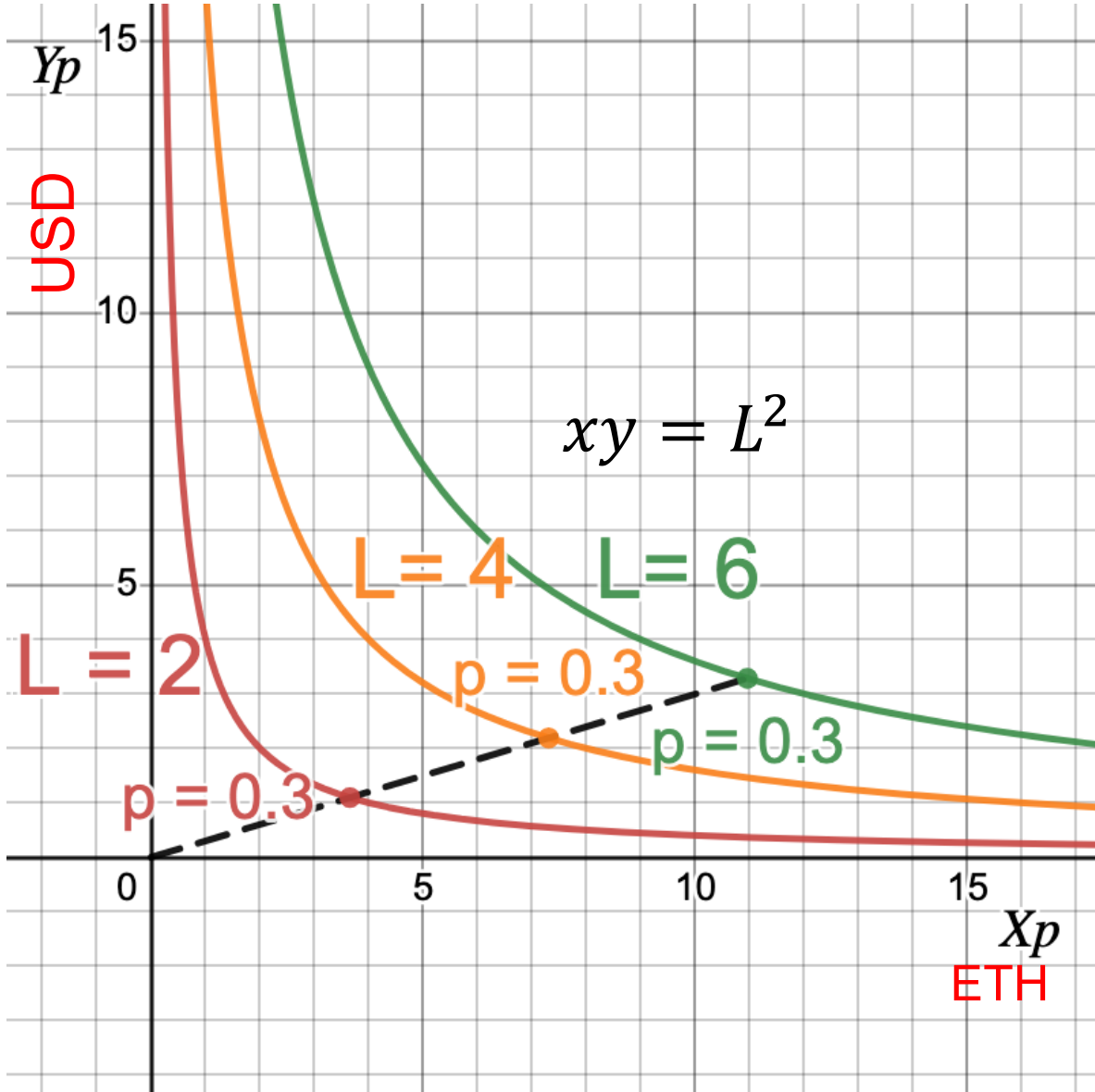
Why is this useful?

- Easy to combine LPs at any price



Switching curves

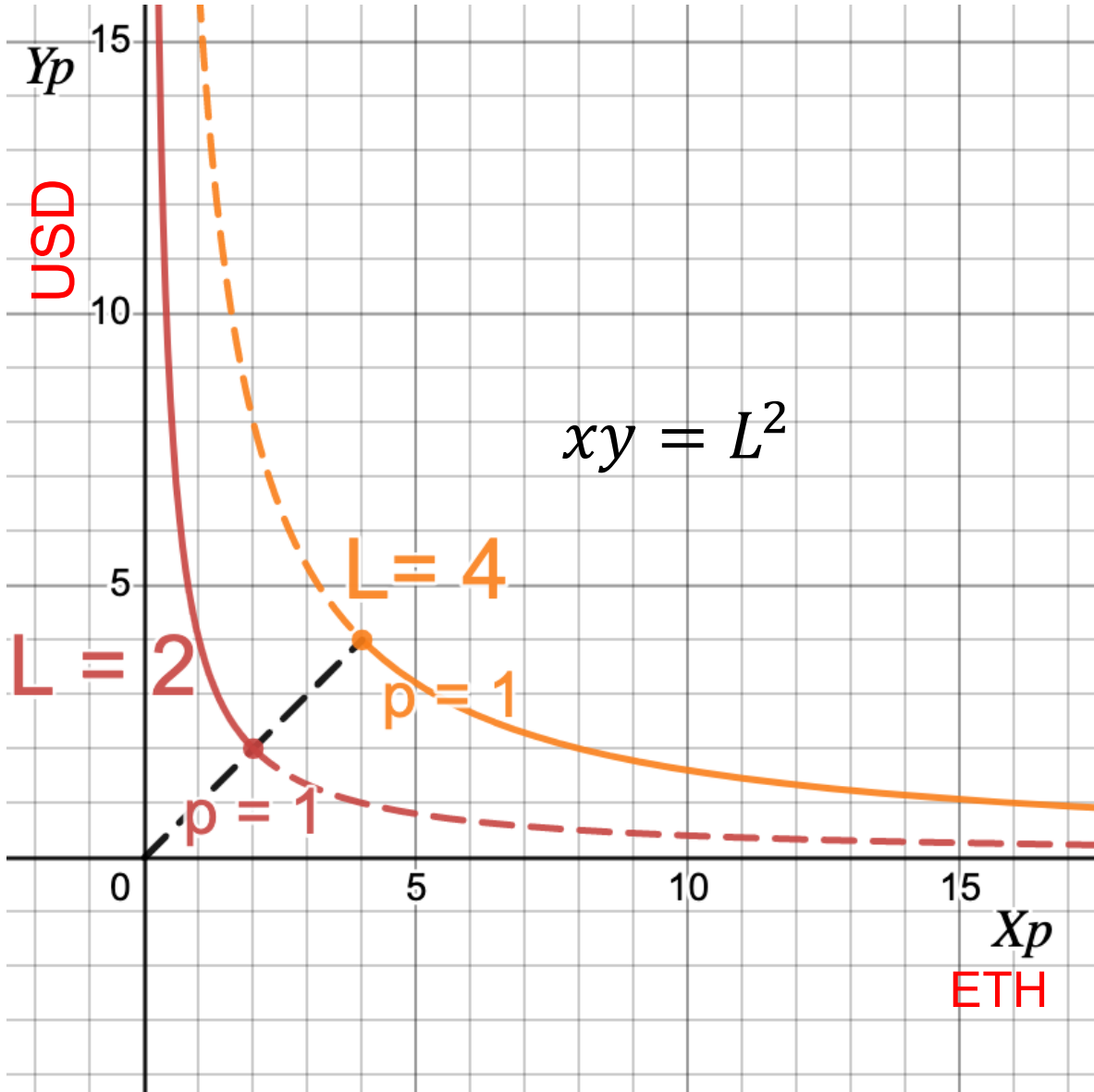
- Need to switch across curves with different liquidity
- CPMMs also make this easy
- Prices along the line through origin are the same – why?
- Switch curves along those lines!



Example 1: Switching

- LP1 only allocates liquidity for price > 1
- LP2 only allocates liquidity for price < 1

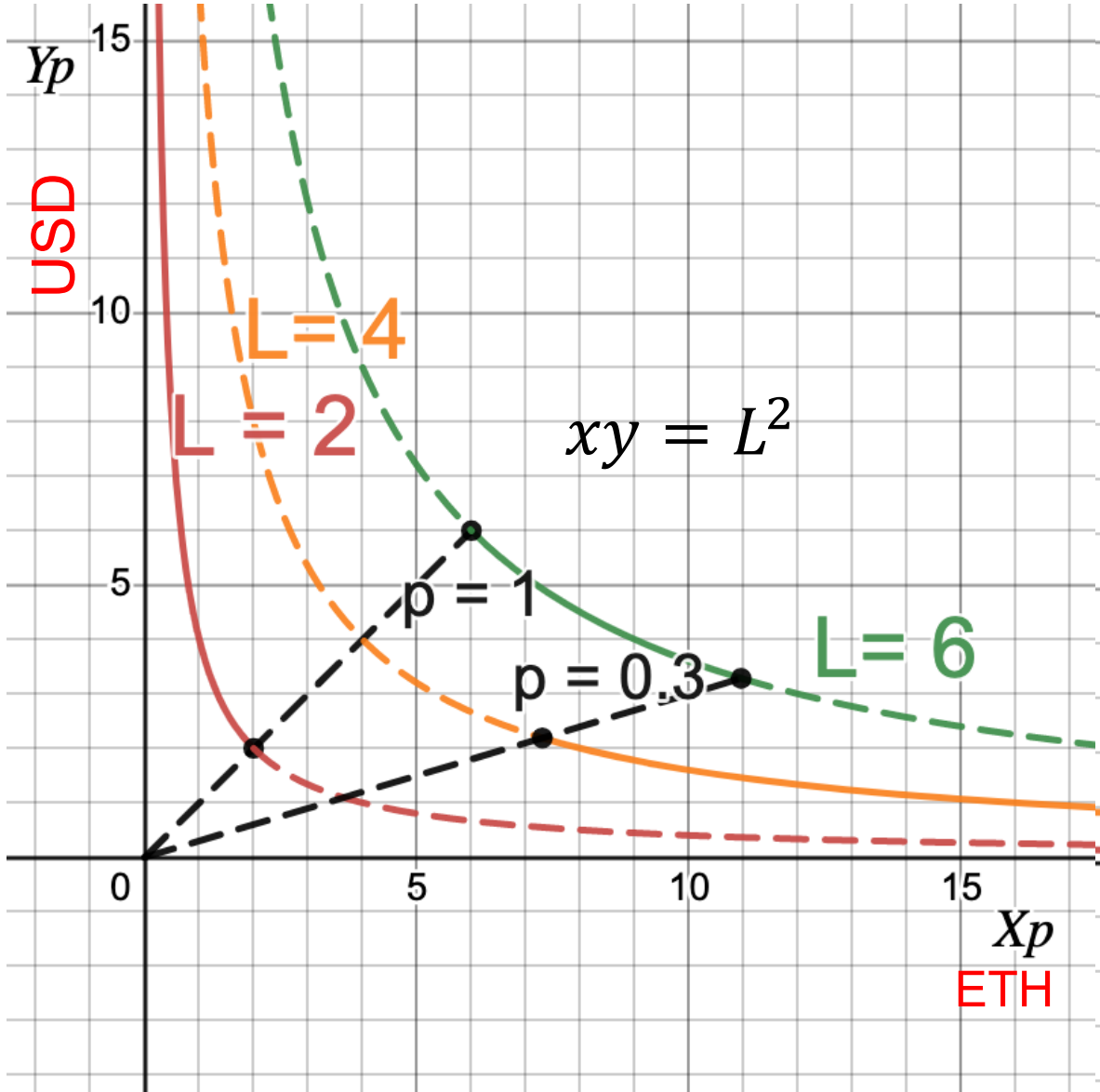
- Reserves only move along the solid curve, switch at p = 1



Example 2: Overlap

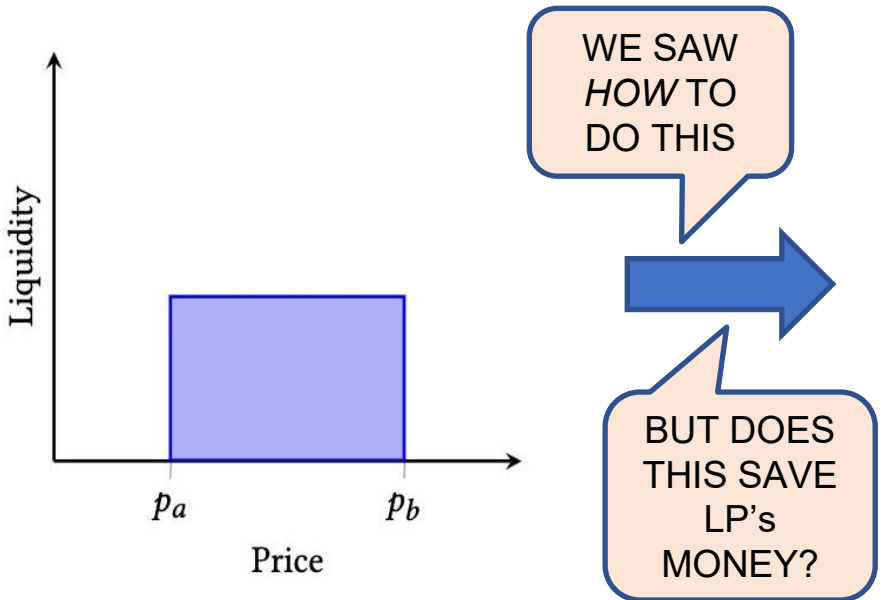
- LP1 only allocates liquidity for price > 0.3
- LP2 only allocates liquidity for price < 1
- What happens when 0.3 < price < 1?

- Reserves only move along the solid curves, switch at p = 0.3 and p = 1



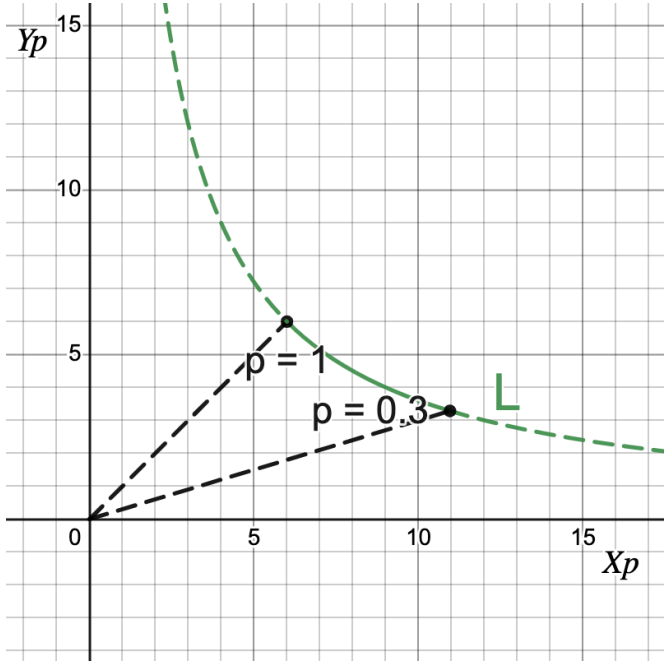
Improving capital efficiency

- So far, we have only looked at the mechanics of how concentrated liquidity would be implemented
- Goal was to improve capital efficiency
- i.e. Are LPs able to enable the same market with the less investment?



WE SAW *HOW* TO DO THIS

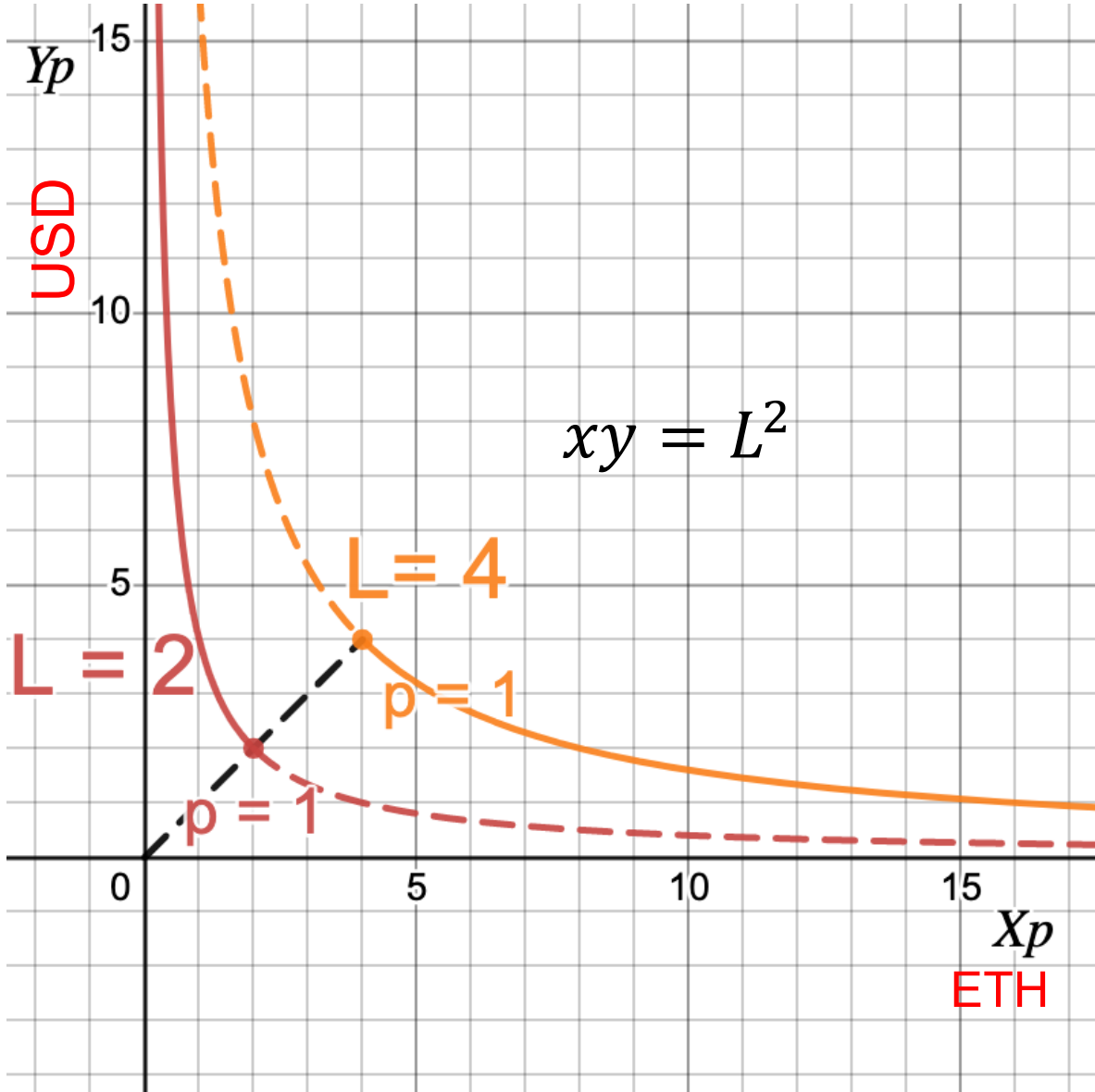
BUT DOES THIS SAVE LP's MONEY?



Improving capital efficiency: example

- LP1 only allocates liquidity for price > 1
- LP2 only allocates liquidity for price < 1
- How much reserves do they need to invest at p = 1?

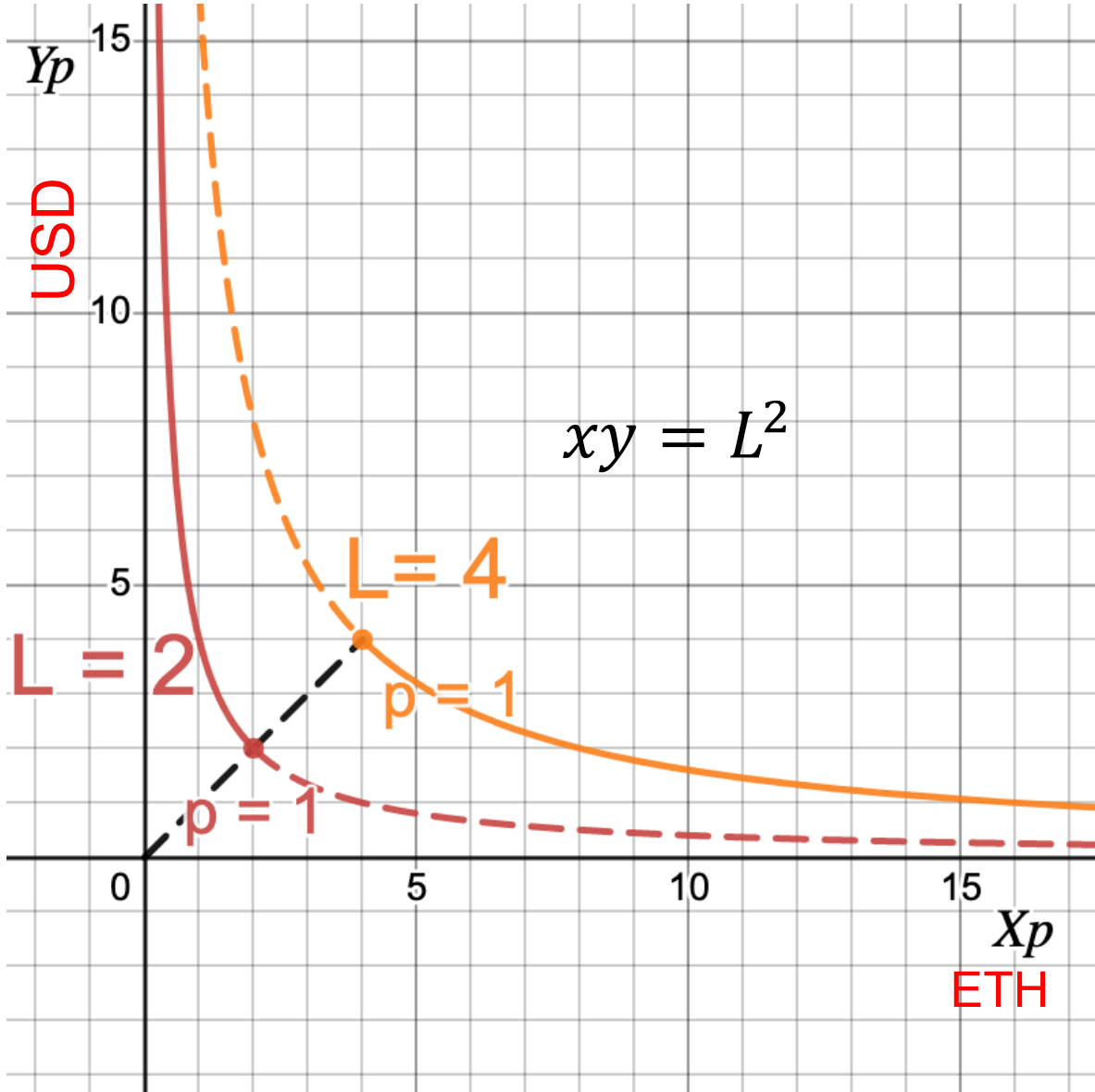
- LP1 normally gives (2 ETH, 2 USD)
- LP2 normally gives (4 ETH, 4 USD)



Improving capital efficiency: example

- LP1 normally gives
  - (2 ETH, 2 USD)
- LP2 normally gives
  - (4 ETH, 4 USD)
- But, LP1 not active when p < 1
- Does not need USD reserves!

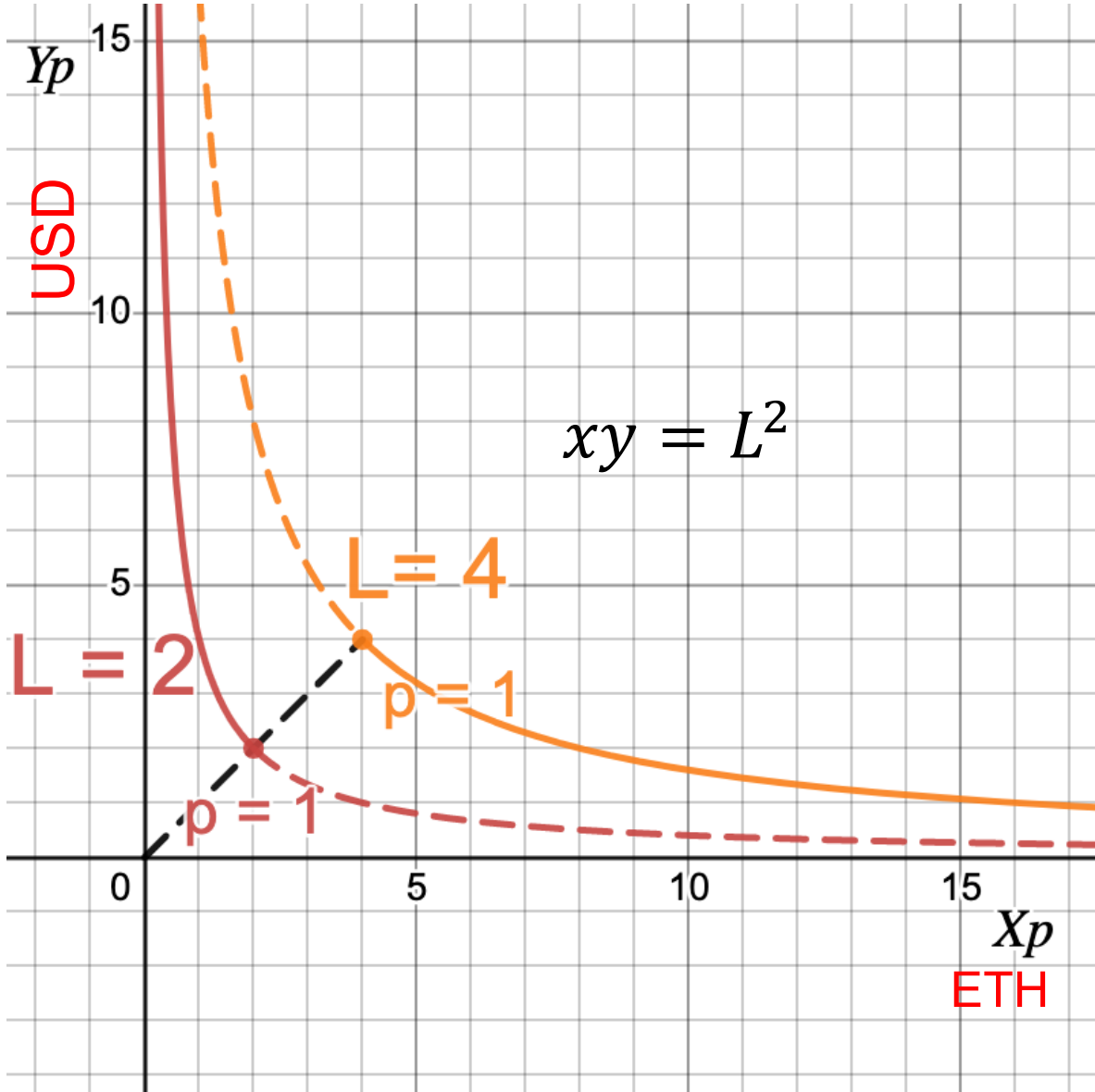
- LP1's real reserves :
  - (2 ETH, 0 USD)
- LP2's real reserves :
  - (0 ETH, 4 USD)



Improving capital efficiency: example

- LP1's real reserves :
  - (2 ETH, 0 USD)
- LP2's real reserves :
  - (0 ETH, 4 USD)

- LP1 follows :
  $$(x)(y+2) = 4$$
- LP2 follows :
  $$(x+4)(y) = 16$$

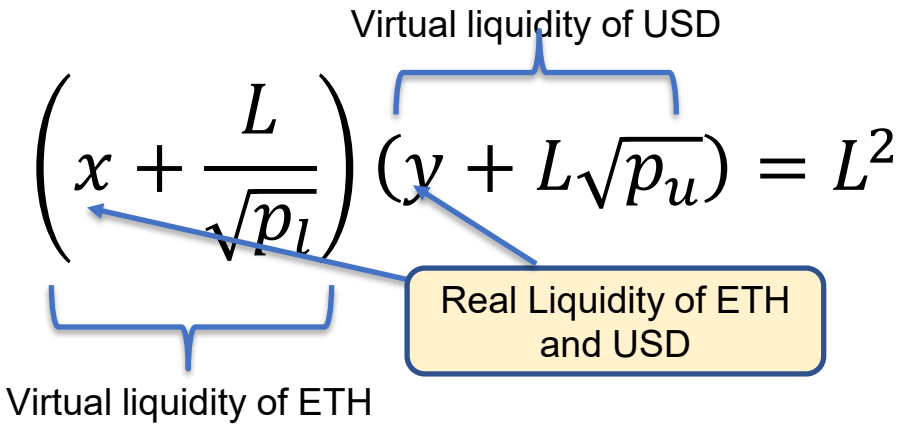


Improving capital efficiency

- Previous example – tells us an LP need only invest a small amount of (ETH,USD) when their chosen price range is smaller
- In practice, LP come in with the (ETH,USD) and a price range
- Smart contracts tells LP their operating curve (value of L)
- For the same investment, LPs get a better curve (larger L) than in simple CPMMs

General Formula

- When LP wants to invest (x, y) between prices p_u and p_l



$$P_x = \frac{y + L\sqrt{p_u}}{x + \frac{L}{\sqrt{p_l}}}$$

- When multiple LPs, simply add their liquidities
- Fees distributed in proportion to liquidities
- What would the price be at (x, y) ? – slope of the curve

Properties: LP's Perspective

Today's Lab - Liquidity Provision in Uniswap v3

$$\left(x + \frac{L}{\sqrt{p_l}}\right)(y + L\sqrt{p_u}) = L^2$$

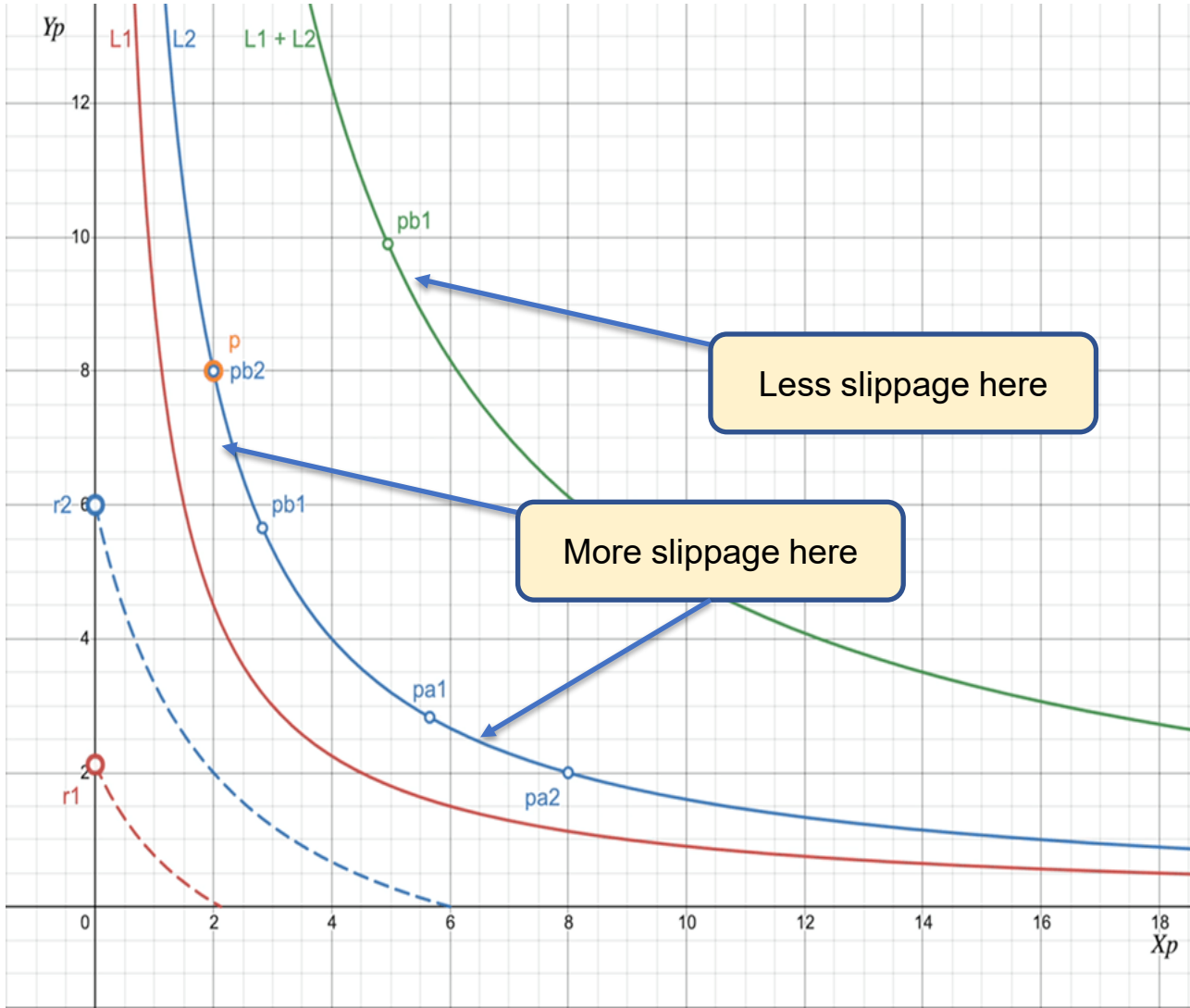
Suppose an LP comes in with wealth (x, y) of (ETH,USD) tokens

- When price range is narrower:
  - For the same investment, liquidity L increases – market depth increases
  - For the same investment, LPs get a larger share of fees
- Cap on impermanent loss
- Tradeoff: investment is inactive (earns no fees) when price outside range – smaller range makes this more likely
- LPs need to keep predicting where price would be in the future to maximize fee revenue

Properties: Trader's Perspective

$$\left(x + \frac{L}{\sqrt{p_l}}\right)(y + L\sqrt{p_u}) = L^2$$

- Trader faces less slippage when market is deeper (large L)
- If LPs allocate liquidity where the price is most likely to be, then traders get a deep market always



Open problems: Concentrated Liquidity

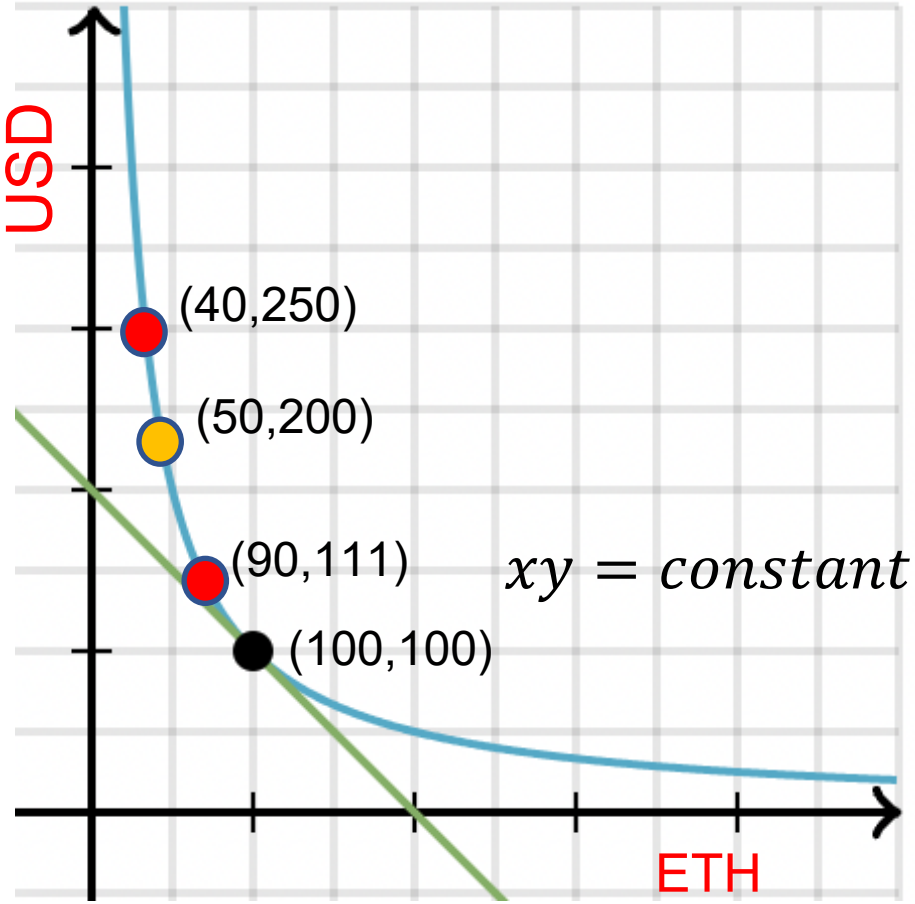
- Best algorithm to move liquidity around?
  - Given the price history, anticipate where price would be
  - Maximize LP profit
    - Note that LPs have to pay gas fees for moving liquidity around – need to balance that with higher share of fees being obtained
- Just-In-Time liquidity:
  - LP sandwiches trades between allocating and pulling out liquidity – atomic!
  - LP gets most of the share of fees
  - Other passive LPs lose out
  - Is this good or bad?
  - Good for traders, Bad for other LPs

Recall: Front Running

MEV : Sandwich Attack

- User wants to do a normal trade :
  - Buy 50 ETH, (has to pay 100 USD normally)

- If miner sees a large buy txn,
  - Introduce a buy txn just before it : buy 10 ETH
  - Put the txn
  - Introduce a sell txn just after it : sell 10 ETH
- Miner gets profit with no risk : 39 USD
- User gets a worse price : 139 USD



Solutions to Front Running

- Main cause behind frontrunning ?
  - Ordering of transactions enforced by third party
  - Transaction Value and Direction (Buy/Sell) is public

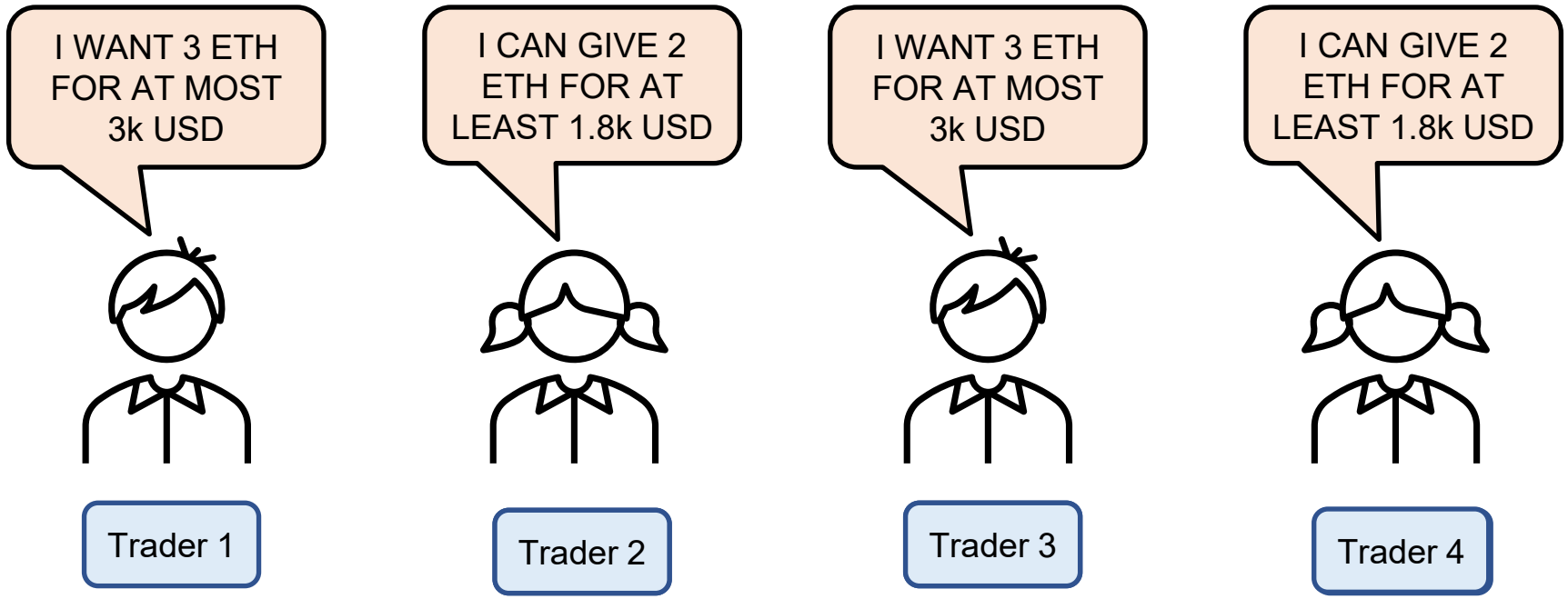
- Make order of transactions irrelevant – how?
- Batch transactions – everyone gets the same price within a batch

- Make transactions opaque – how?
- Private CFMMs – no one can see contents of transaction

Batching trades

- Collect all trades within a block into a batch
- Compute a uniform clearing price
- All trades executed at the same price

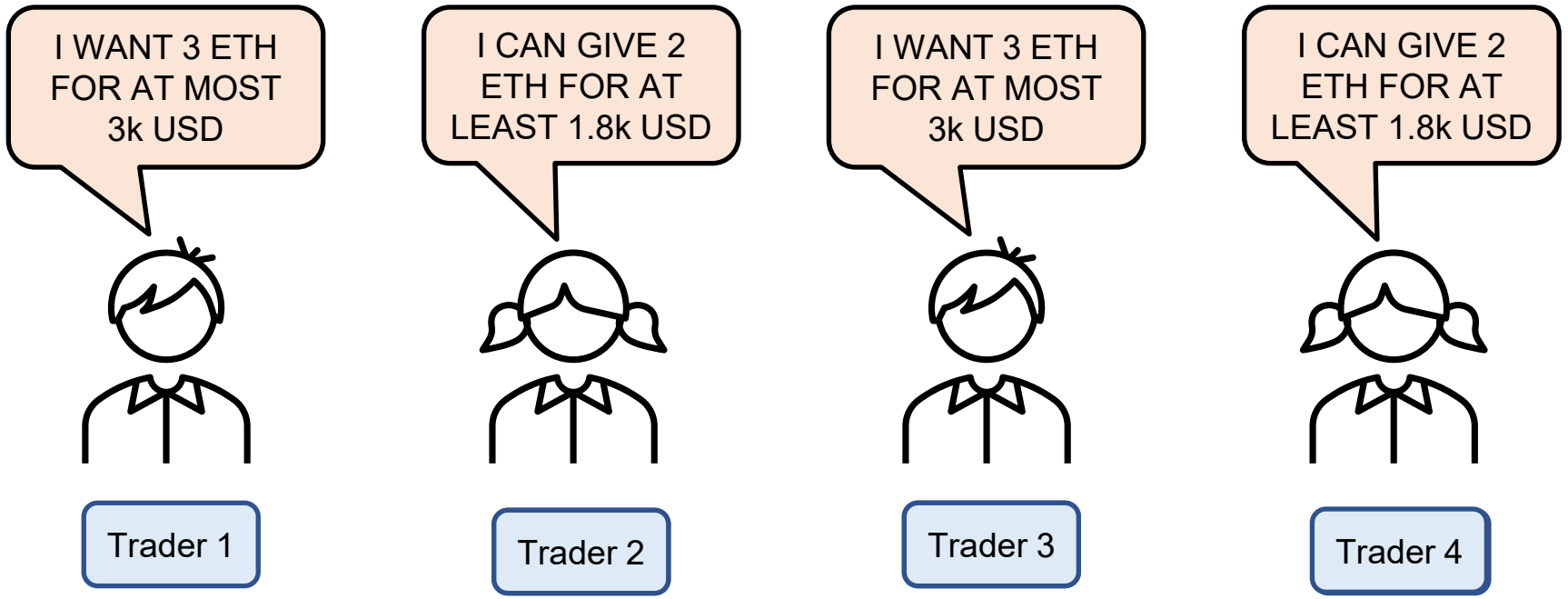
ENFORCING THIS GETS RID OF FRONTRUNNING AND ARBITRAGE OPPORTUNITIES



Batching trades

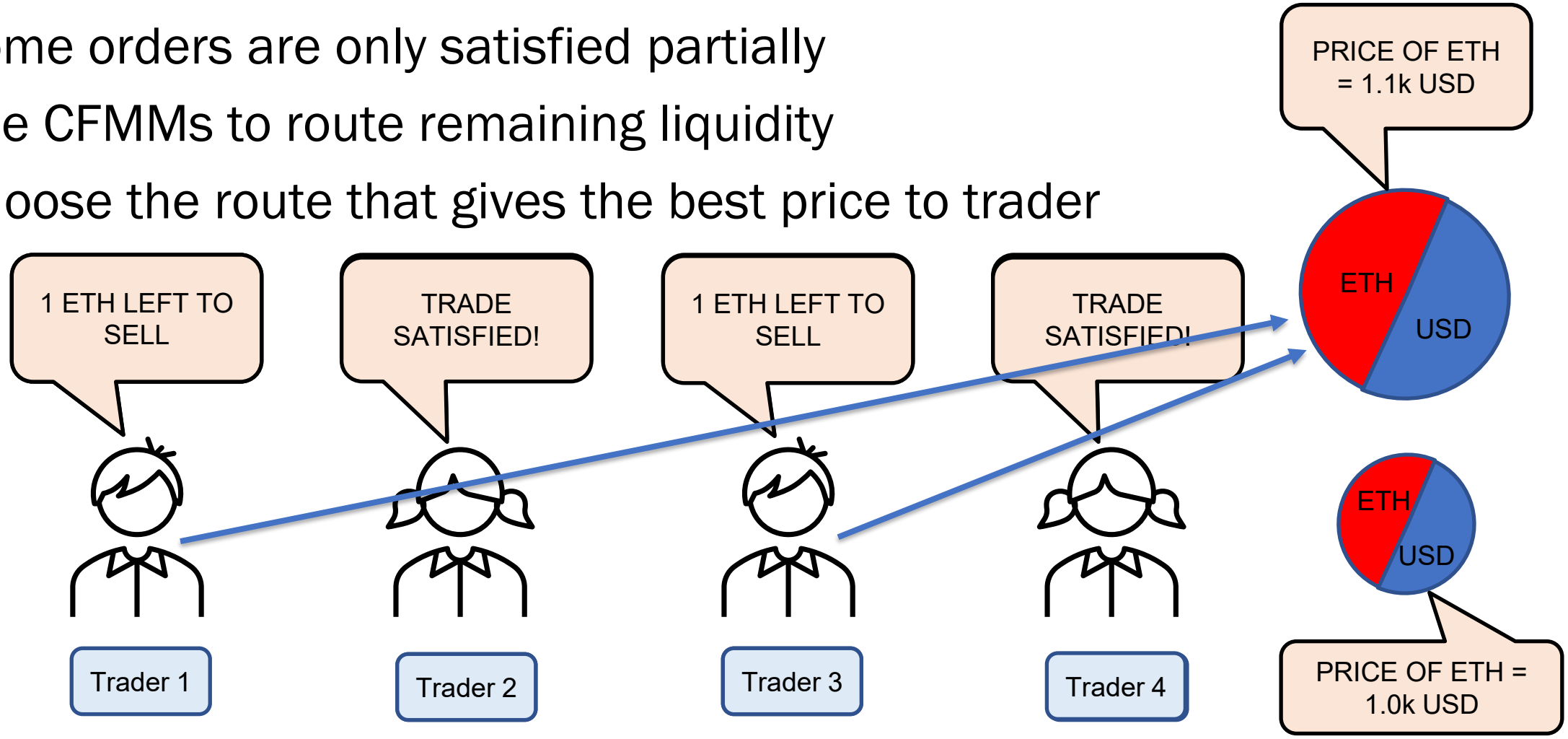
- What would be a fair clearing price be in this case?
- Buyers not willing to go above 1k USD/ETH
- Sellers not willing to go below 0.9k USD/ETH

Clearing price = 0.95k



Batching trades

- Some orders are only satisfied partially
- Use CFMMs to route remaining liquidity
- Choose the route that gives the best price to trader



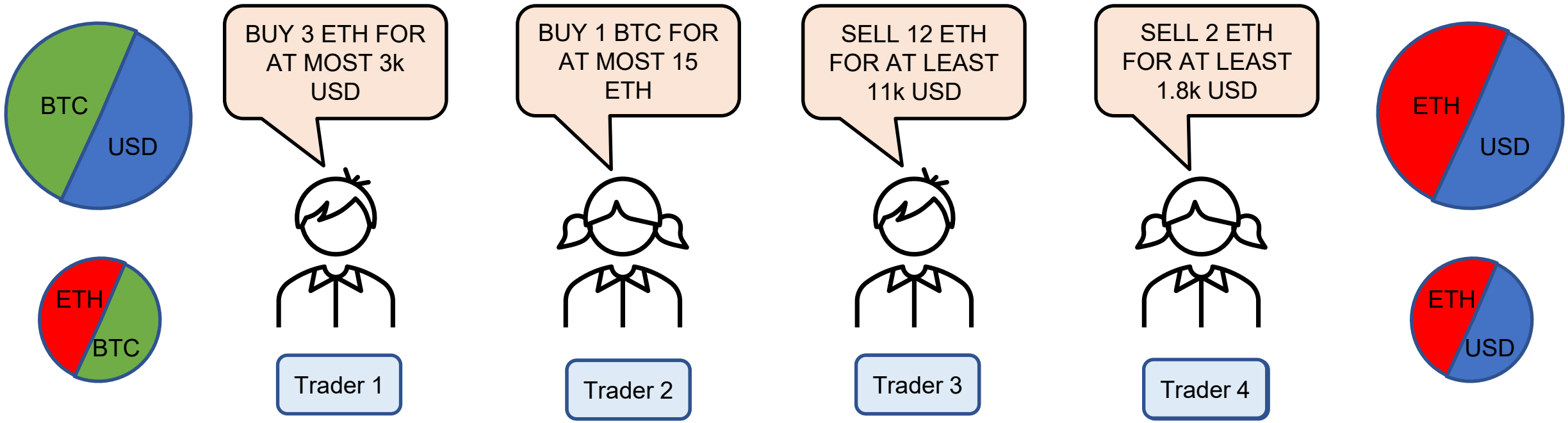
Batching trades: the general case

- Trades between different pairs of tokens?
- Batch and compute a clearing price
- Trades with the same pair should be executed at the same price

COMPUTING SUCH CLEARING PRICES IS NP-HARD IN GENERAL

CAN USE APPROXIMATIONS

GETS RID OF FRONTRUNNING AND MULTI-HOP ARBITRAGE OPPORTUNITIES



Private CFMMs

- Provide exchange services privately
- ZK cryptography
- Example: Penumbra

LECTURE ENDS