

CPSC 3640/5400: Decentralized Payments, Contracts, and Finance for Humans and AI

Lecture 12: AMM Part 2

Fan Zhang
Yale University
2026 Fall

Recap

- LOB v.s. AMM
- Traders v.s. Liquidity providers
- Bonding curve
- CP-MM $xy = k$
- Spot price
- Slippage

What if price drifts from fair price?

- Price on AMM changes as traders trade
- Creating an **arbitrage** opportunity: riskless profit obtained by exploiting price differences

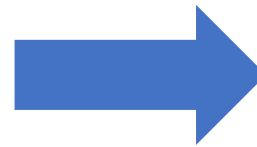
ETH trading at 3000 USD

 **UNISWAP**

Buy low



Sell high



ETH trading at 3050 USD



What if price drifts from fair price?

- Due to arb, ETH price rises on Uniswap and lowers on Binance
- When should the **arbitrageur** stop?
- Arbitrageurs help keep AMM price **in sync** with other markets.

ETH trading at 3000 USD

 **UNISWAP**

Buy low



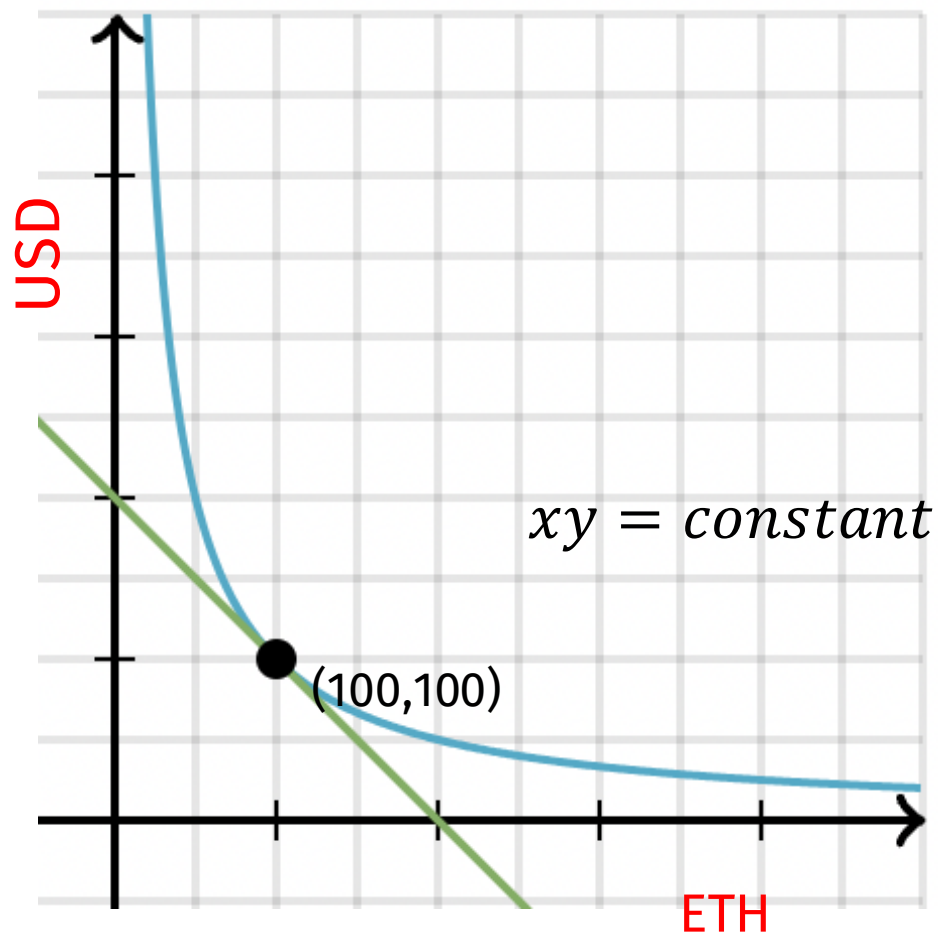
Sell high



ETH trading at 3050 USD

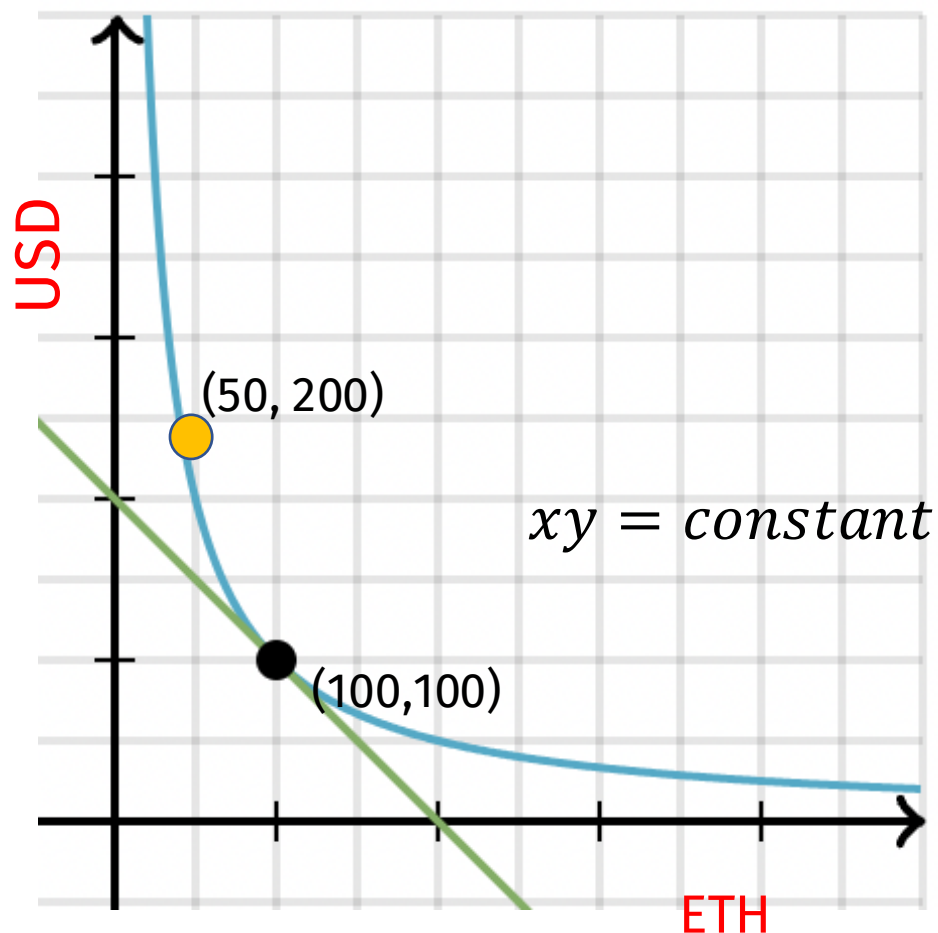


Arbitrage (calculation)



- Spot price at $p = 1$
- ETH price jumps to $p' = 4$ outside AMM
- How should **arbitrageur** trade?
- Introducing the **demand curve**
- $p_x = \frac{y}{x} \Rightarrow x(p) = \sqrt{k/p_x}$
- Similarly, $y(p) = \sqrt{k p_x}$

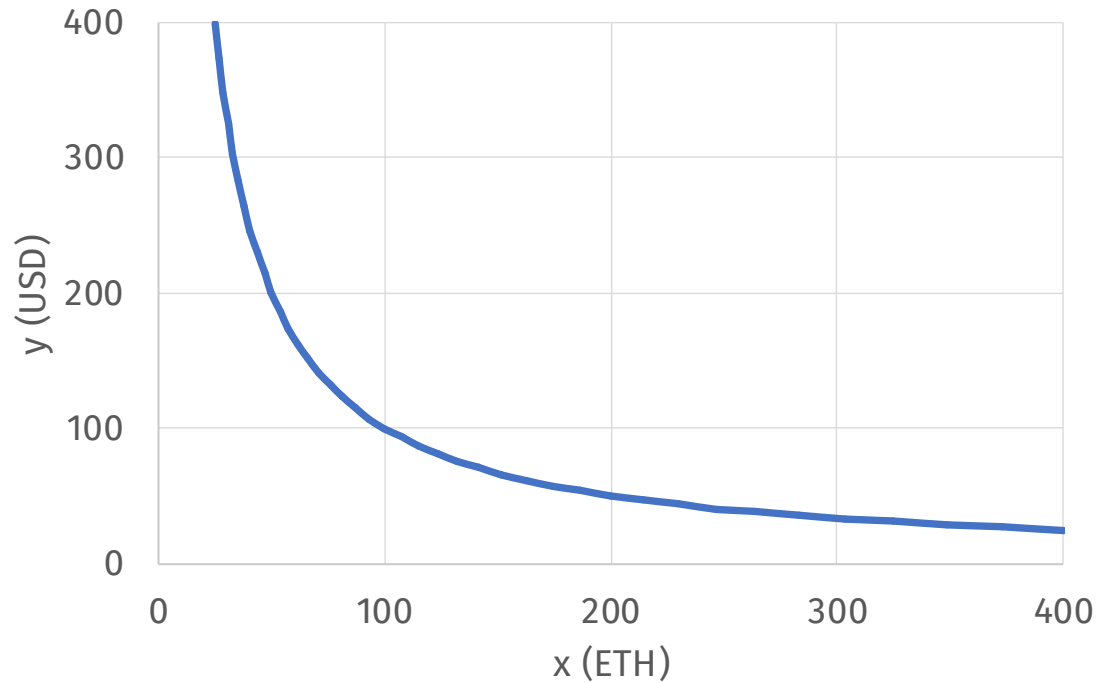
Arbitrage (calculation)



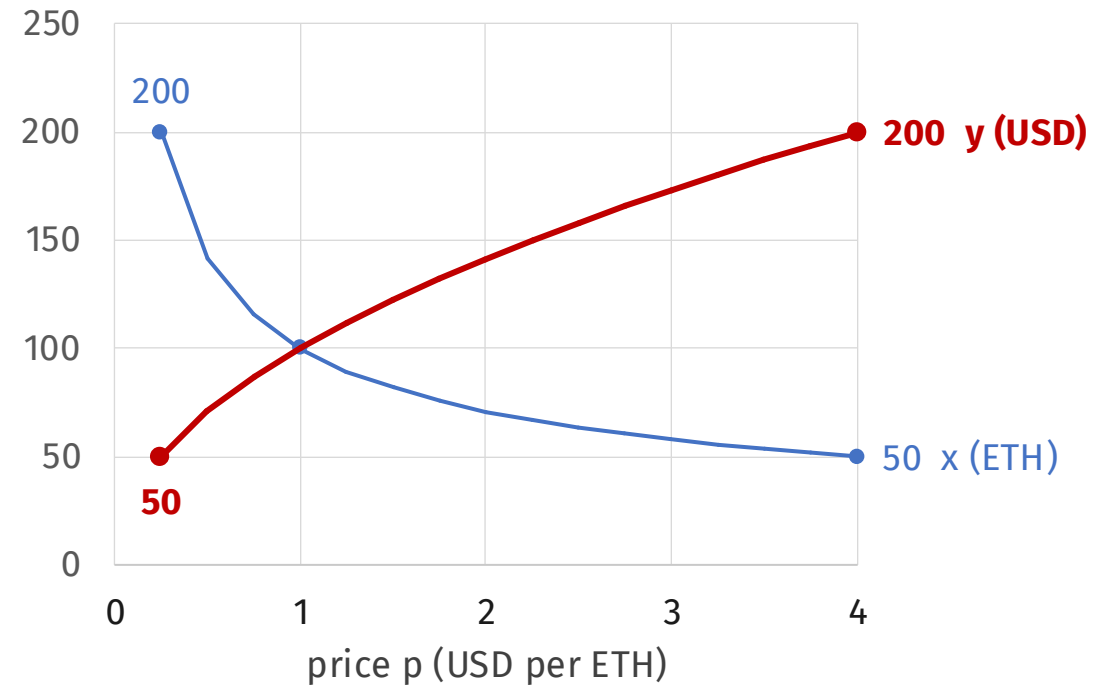
- Spot price at $p = 1$
- ETH price jumps to $p' = 4$ outside AMM
- How should **arbitrageur** trade?
 - After arbitrage $x = 100/\sqrt{4} = 50$
 - The trade is to buy 50 ETH & sell outside AMM
- What is the arbitrage profit?
 - Cost: 100
 - Revenue: $50 * 4$
 - Profit: 100 USD

Demand curve

Reserve view: $xy = 10,000$



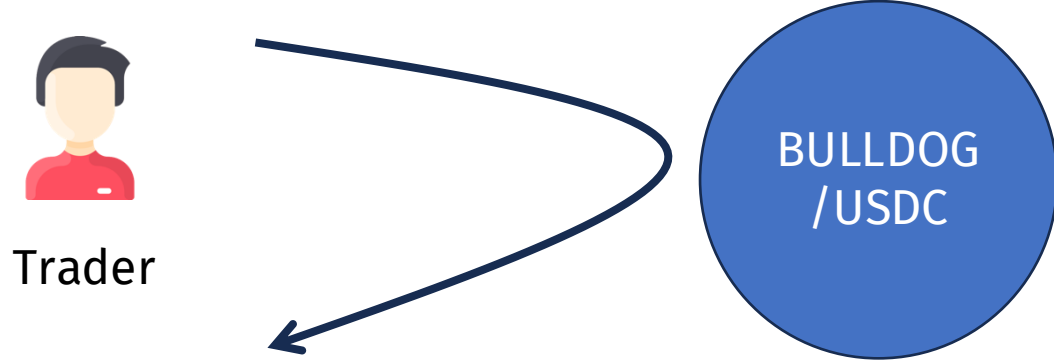
Price view: $x = \frac{100}{\sqrt{p}}$, $y = 100\sqrt{p}$



- Handy for computation involving prices

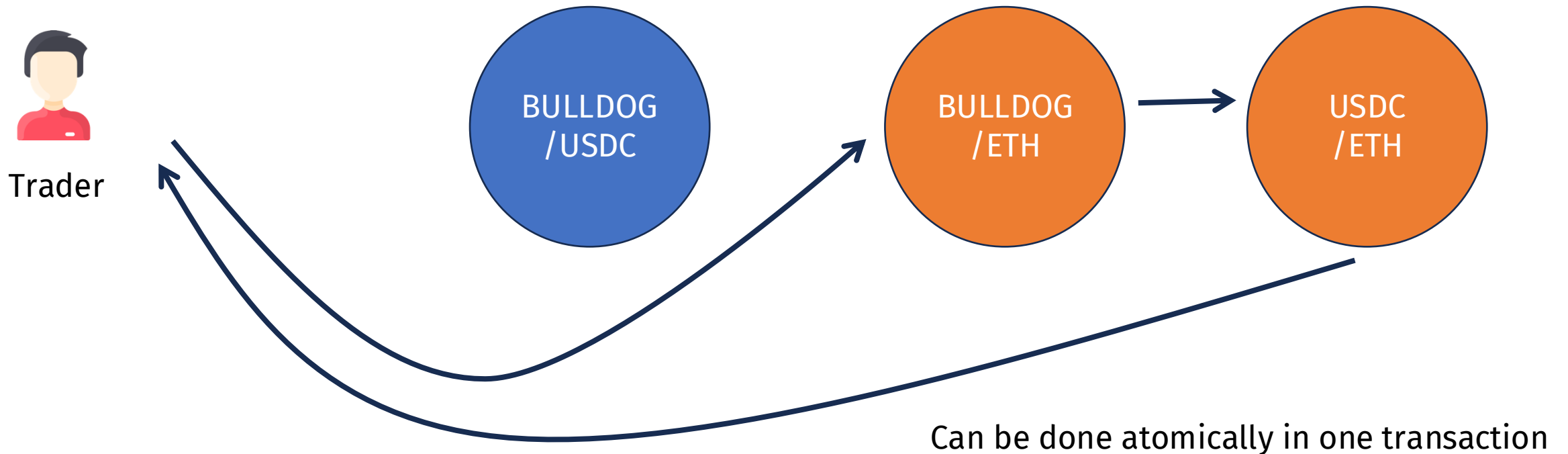
Routing

- Want to sell 10 BULLDOGS for USDC
- If there is one pool, then we don't have other choices



Routing

- Want to sell 10 BULLDOGS for USDC
- We also have BULLDOG/ETH
- Now a second path is possible BULLDOGS -> ETH -> USDC



Optimal Routing for Constant Function Market Makers

GUILLERMO ANGERIS, Stanford University

ALEX EVANS, Bain Capital Crypto

TARUN CHITRA, Gauntlet Networks

STEPHEN BOYD, Stanford University

We consider the problem of optimally executing an order involving multiple crypto-assets, sometimes called tokens, on a network of multiple constant function market makers (CFMMs). When we ignore the fixed cost associated with executing an order on a CFMM, this optimal routing problem can be cast as a convex optimization problem, which is computationally tractable. When we include the fixed costs, the optimal routing problem is a mixed-integer convex problem, which can be solved using (sometimes slow) global optimization methods, or approximately solved using various heuristics based on convex optimization. The optimal routing problem includes as a special case the problem of identifying an arbitrage present in a network of CFMMs, or certifying that none exists.

CCS Concepts: • **Mathematics of computing** → **Convex optimization**; • **Applied computing** → **Operations research**.

Additional Key Words and Phrases: convex optimization, optimal routing, decentralized finance, blockchains

ACM Reference Format:

Guillermo Angeris, Alex Evans, Tarun Chitra, and Stephen Boyd. 2022. Optimal Routing for Constant Function Market Makers. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC '22), July 11–15, 2022, Boulder, CO, USA*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3490486.3538336>

Demo a swap

- BULLDOGS <> USDC Pool:
https://app.uniswap.org/explore/pools/ethereum_sepolia/0x0c6219eEe050EA520c7FFc66292f6C7A8A961002
- Do a swap
- See how routing finds a better price.

AMMs for LPs

1) Loss-vs-hold and 2) loss-vs-arbitrage

Liquidity providers

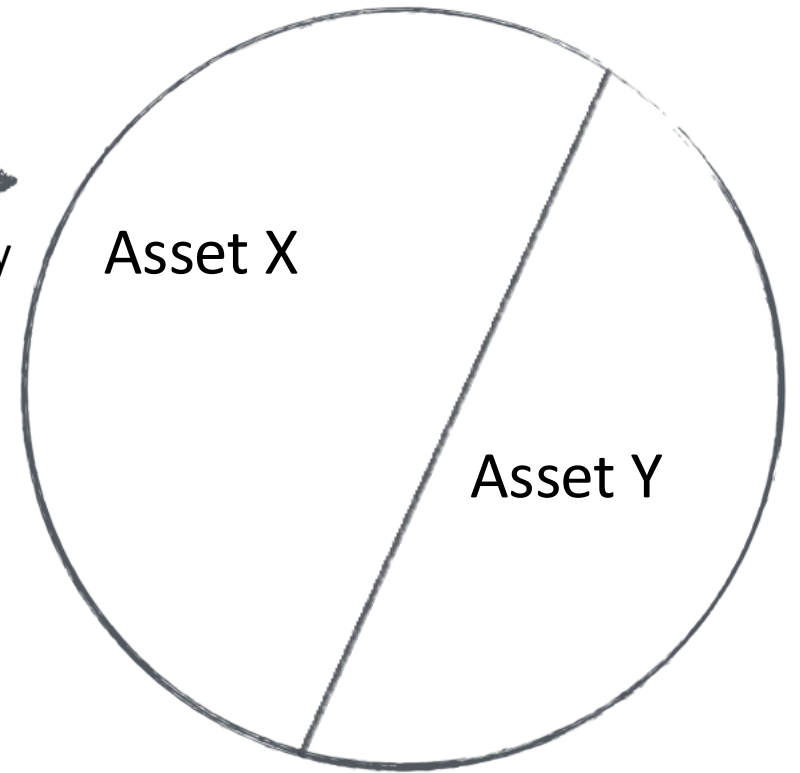
- Providing liquidity: lending money to AMMs for a fee.
- Must provide two assets simultaneously to keep price unchanged:

$$\frac{x}{y} = \frac{x + \Delta x}{y + \Delta y}$$

- Can withdraw at any time.



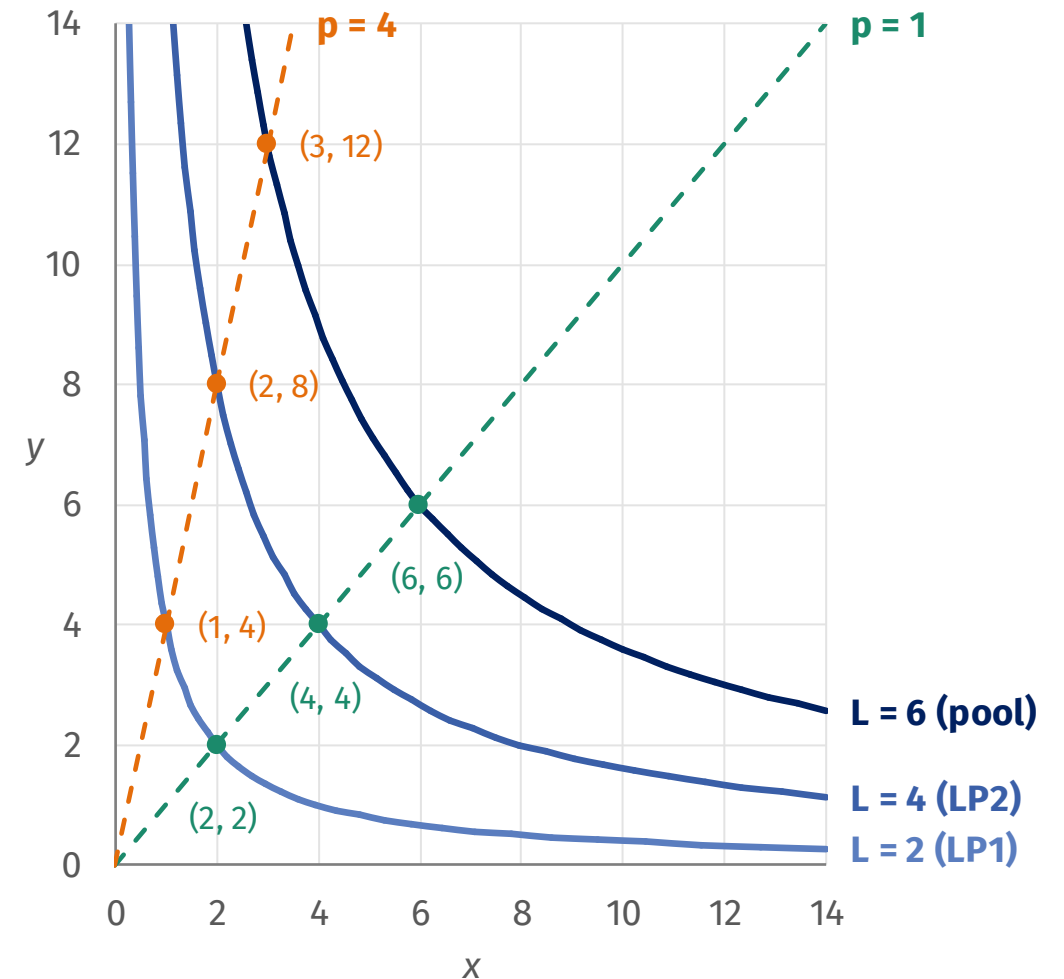
→
Add liquidity
X and Y



Liquidity Pool
(managed by a smart contract)

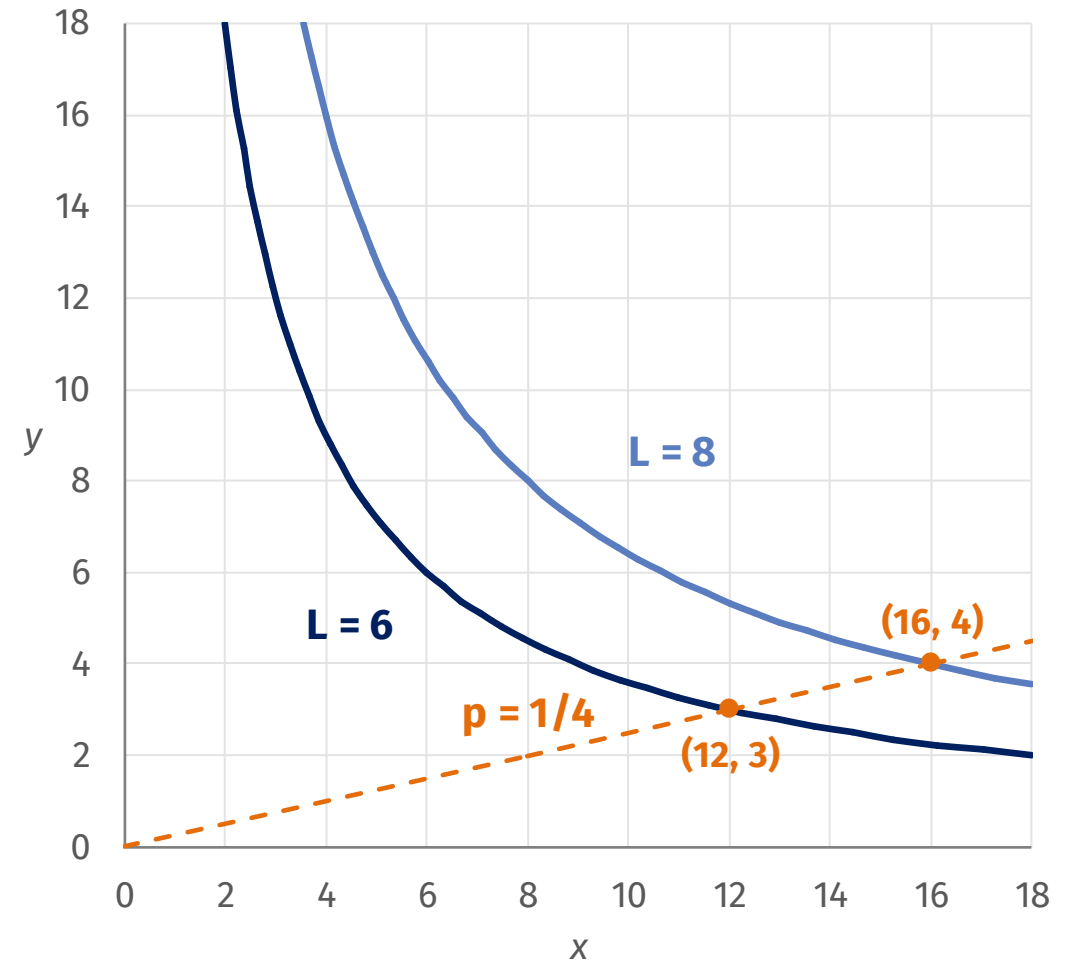
Measure liquidity

- We need a convenient metric to measure liquidity
- We use $L = \sqrt{xy}$ to measure the liquidity of the pool.
- L is additive
- If LP_1 owns $1/3$ of the pool $(x/3, y/3)$, her liquidity is $L_{LP1} = L/3$



Adding more removing liquidity

- At price p , add/removing = switch curves along $y = px$!
- Say we are at $L=6$ and $p = 1/4$, and a user wants to add $L=2$, she needs to add $(4, 1)$.

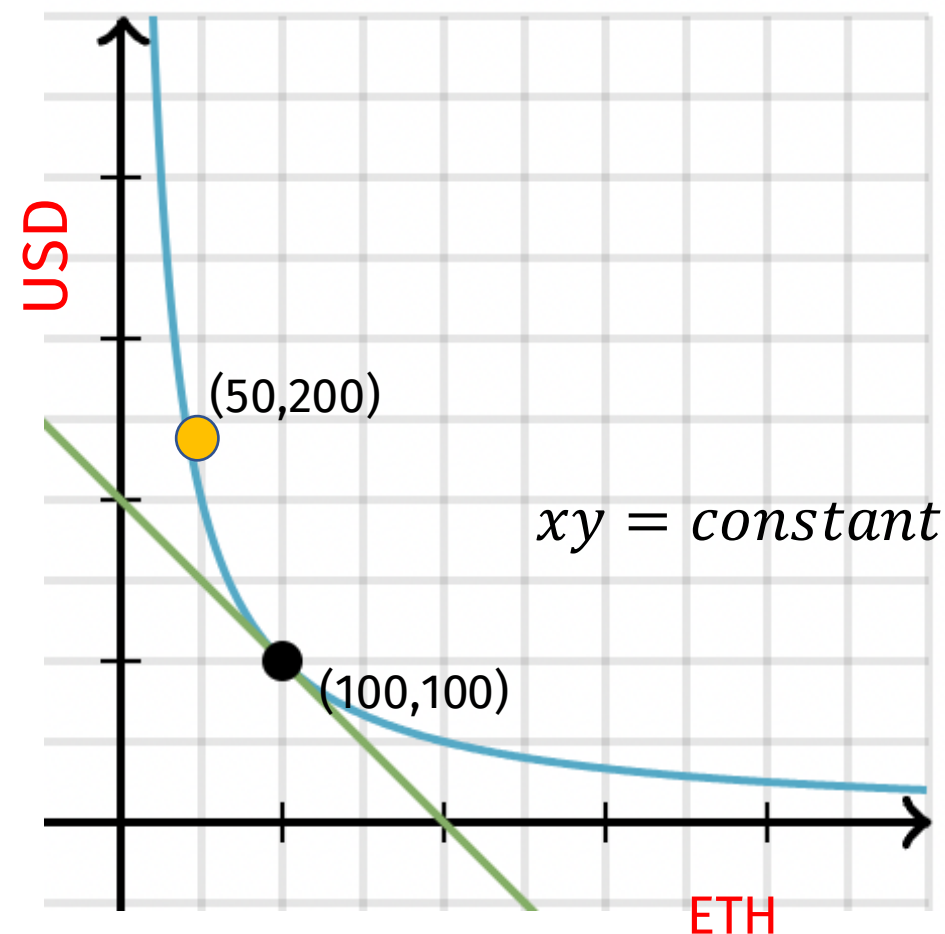


LP's risks

- If asset price changes, what risks does a LP incur?
- Loss vs holding static
- Loss vs arbitrage

Impermanent Loss (Loss-Vs-Holding)

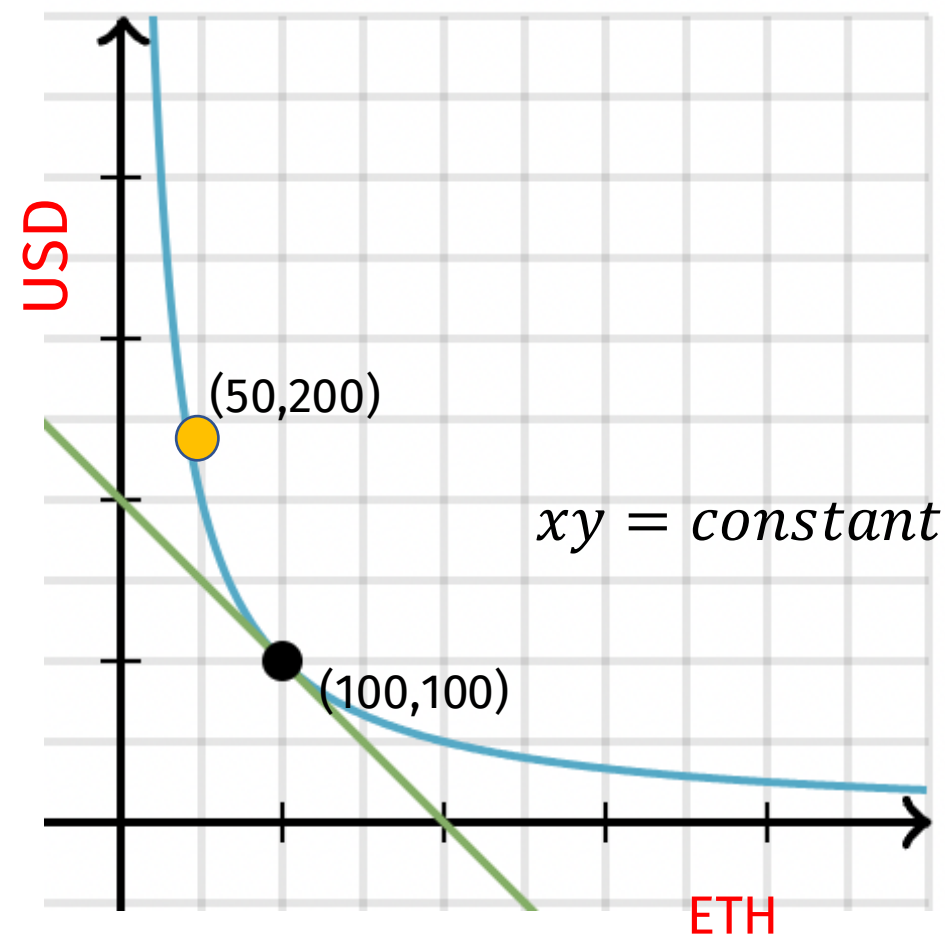
- Looking from the LP's perspective :
- What happens when price goes from 1 USD to 4 USD?
- Value of reserves = $200 + 4 \times 50$ USD
- If held static: $100 + 4 \times 100$ USD
- **Loss = \$100**
- What happens when price goes back to 1 USD?



Impermanent Loss (Loss-Vs-Holding)

Impermanent Loss :

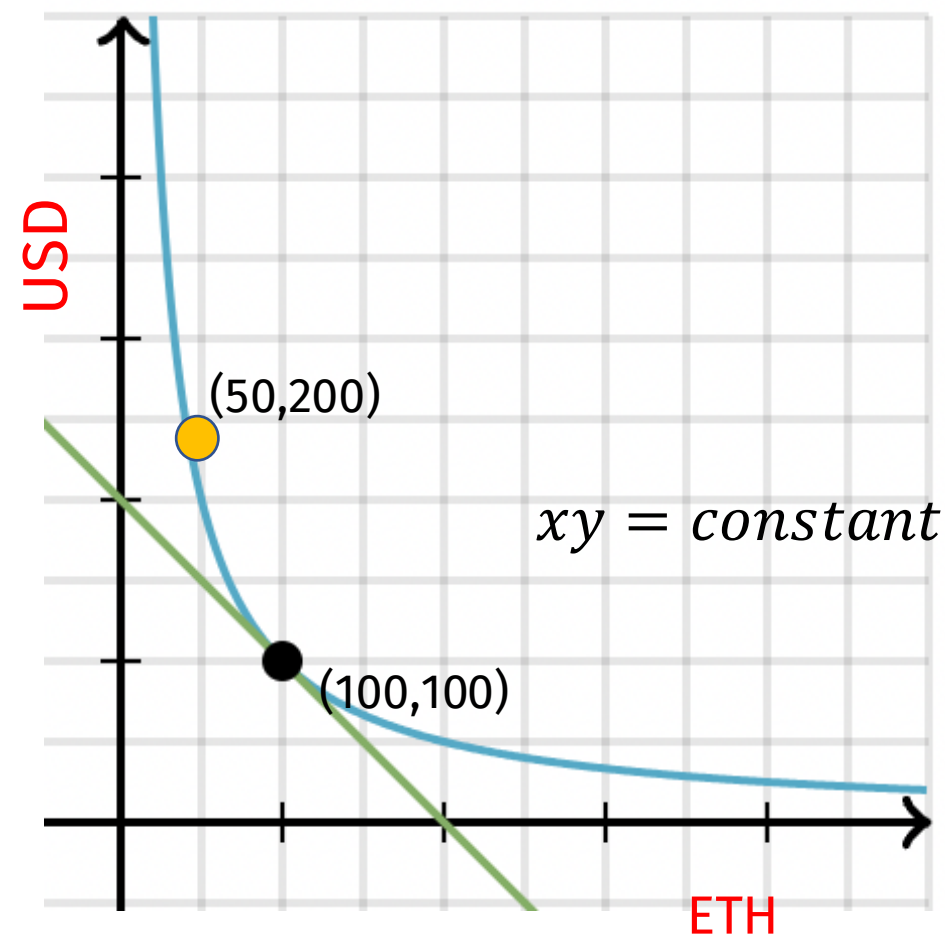
- Value of reserves if held static – Value of reserves under CFMM
- Always positive! **Why?**
- CFMM always sells off the token appreciating in value
- You can also see this using CPMM demand curve formula.



Impermanent Loss (Loss-Vs-Holding)

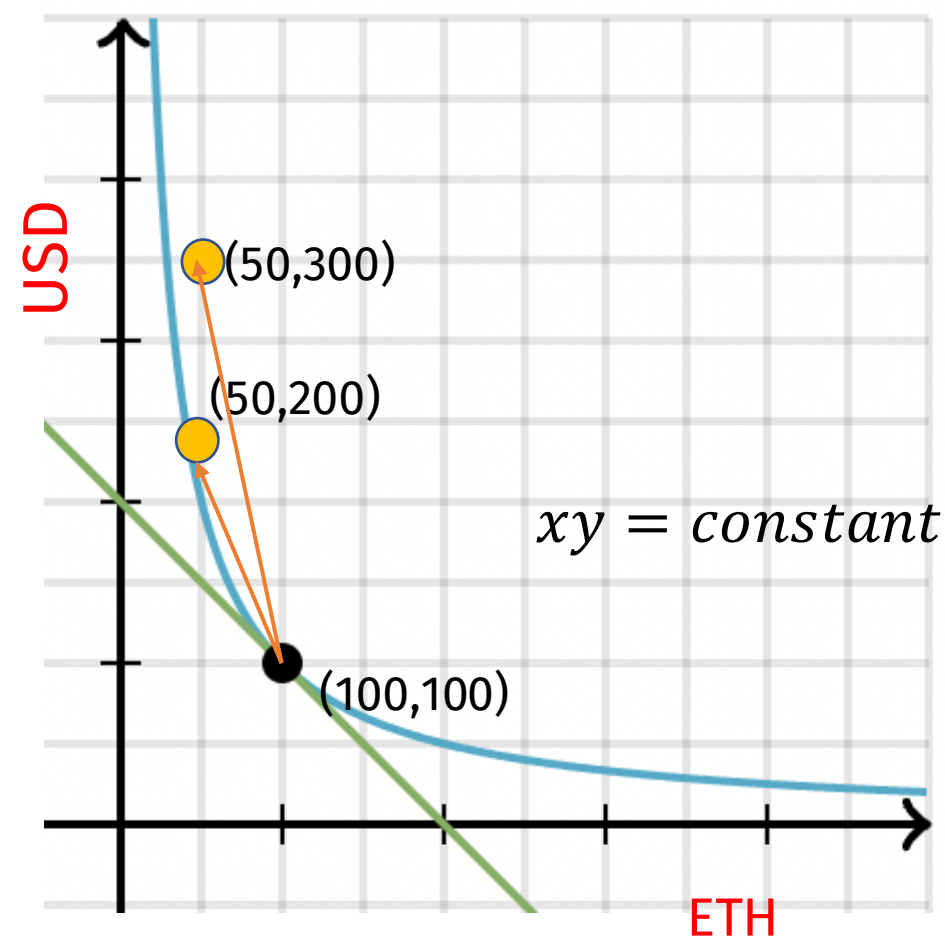
Impermanent Loss :

- Loss disappears when price moves back to the reference point – impermanent
- What is the bet that LPs are making ?



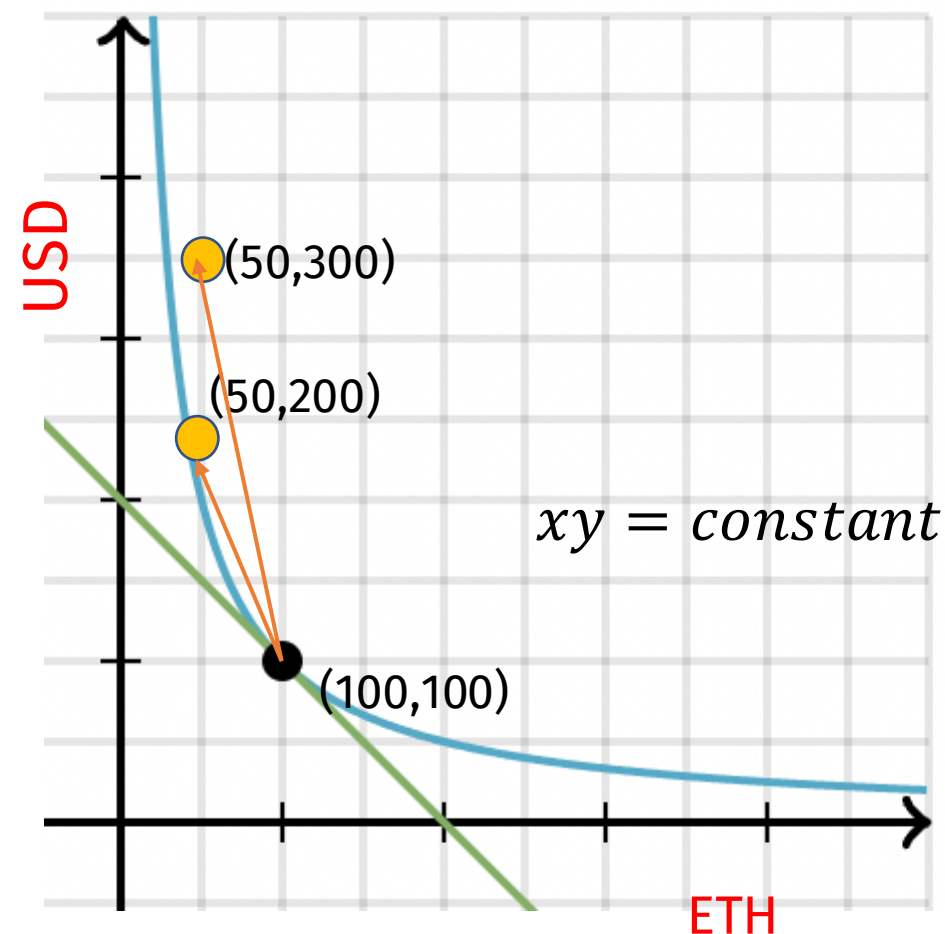
Arbitrage Loss (Loss-Vs-Rebalancing)

- Same situation as before – price jumps from 1 USD to 4 USD
- AMM will sell 50 ETH for 100 USD
- What if sold at “real” price?
- New reserve (50, 300)
- Loss due to arbitrage = 100 USD



Arbitrage Loss (Loss-Vs-Rebalancing)

- Value of trade assuming external price
– Value of trade assuming AMM price
- Exactly the **profit made by the arbitrageur**
- Is non-recoverable and cumulative
- Say price jumps back to 1, AMM buys 50 ETH for 100 USD. Overspent by 50 USD.



Difference between the two losses

Impermanent Loss

- Is path-independent
- Recovered when price reaches initial point again
- Can be **hedged** against
- Depends only on final and initial prices

Arbitrage Loss

- Is path-dependent
- Cannot be recovered
- Need fees to make up for loss
- Depends external price volatility

Fees

- In exchange of allocating liquidity for trades, LPs take fee from asset **coming in**

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

$$Fee = ((1 - \gamma)\Delta_x)$$

- Typically, $\gamma \approx 0.99$

- Induces an effective ask and a bid price

$$p^{ask} = P_x / \gamma$$

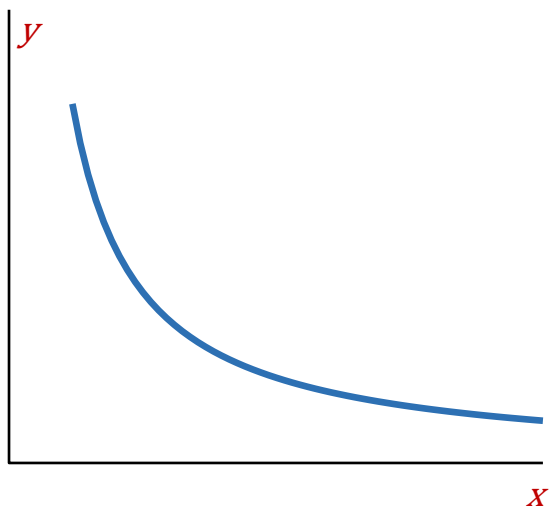
$$p^{bid} = \gamma P_x$$

EXERCISE:
PROVE
THIS

**Improving LP's experience
through concentrated
liquidity (Uniswap v3)**

Three views of the same AMM

Bonding curve



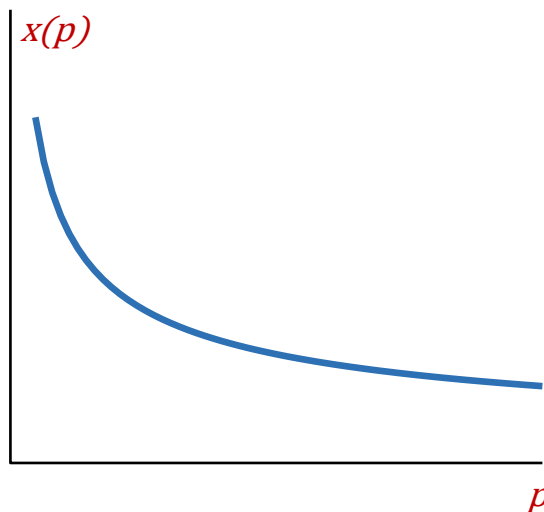
$$\psi(x, y) = L$$

Constrains reserves in (x, y) space

write $x(p)$



Demand curve



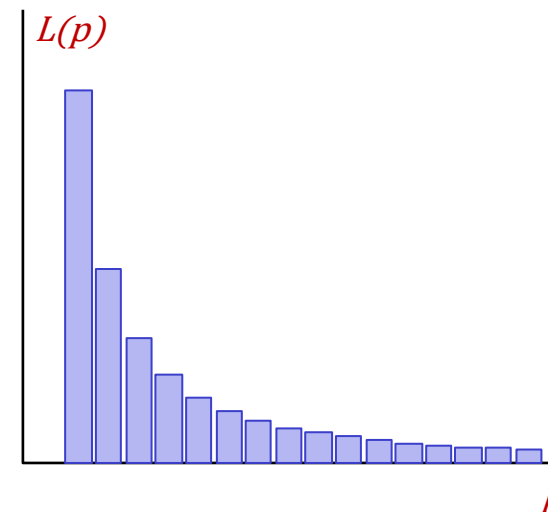
$$x(p)$$

Minimum asset the LP holds at price p

differentiate



Liquidity curve



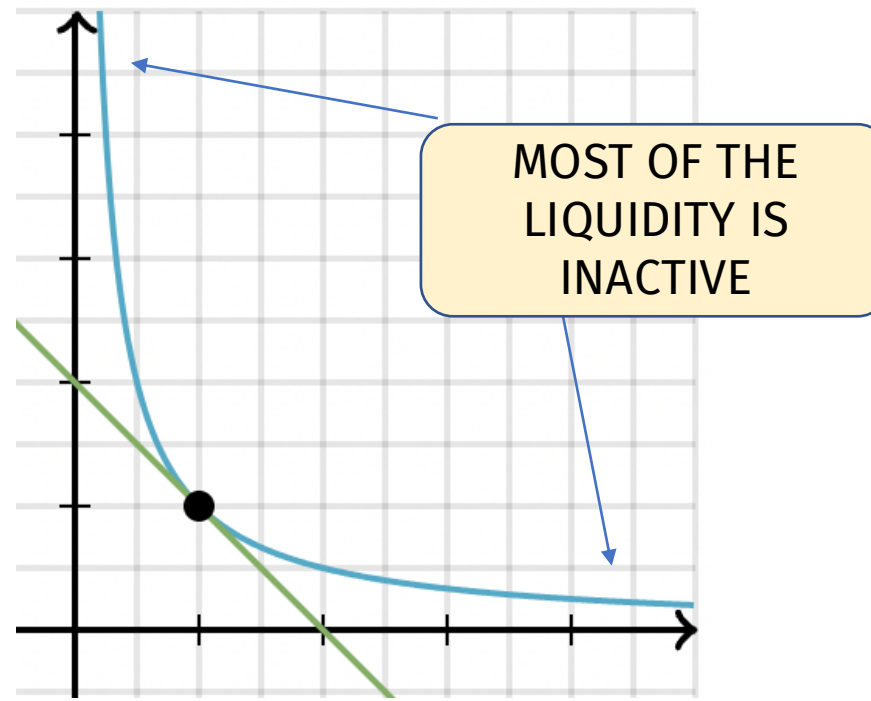
$$L(p) = -x'(p)$$

Asset sold per price step dp

The third graph tell us liquidity spreads across all price range

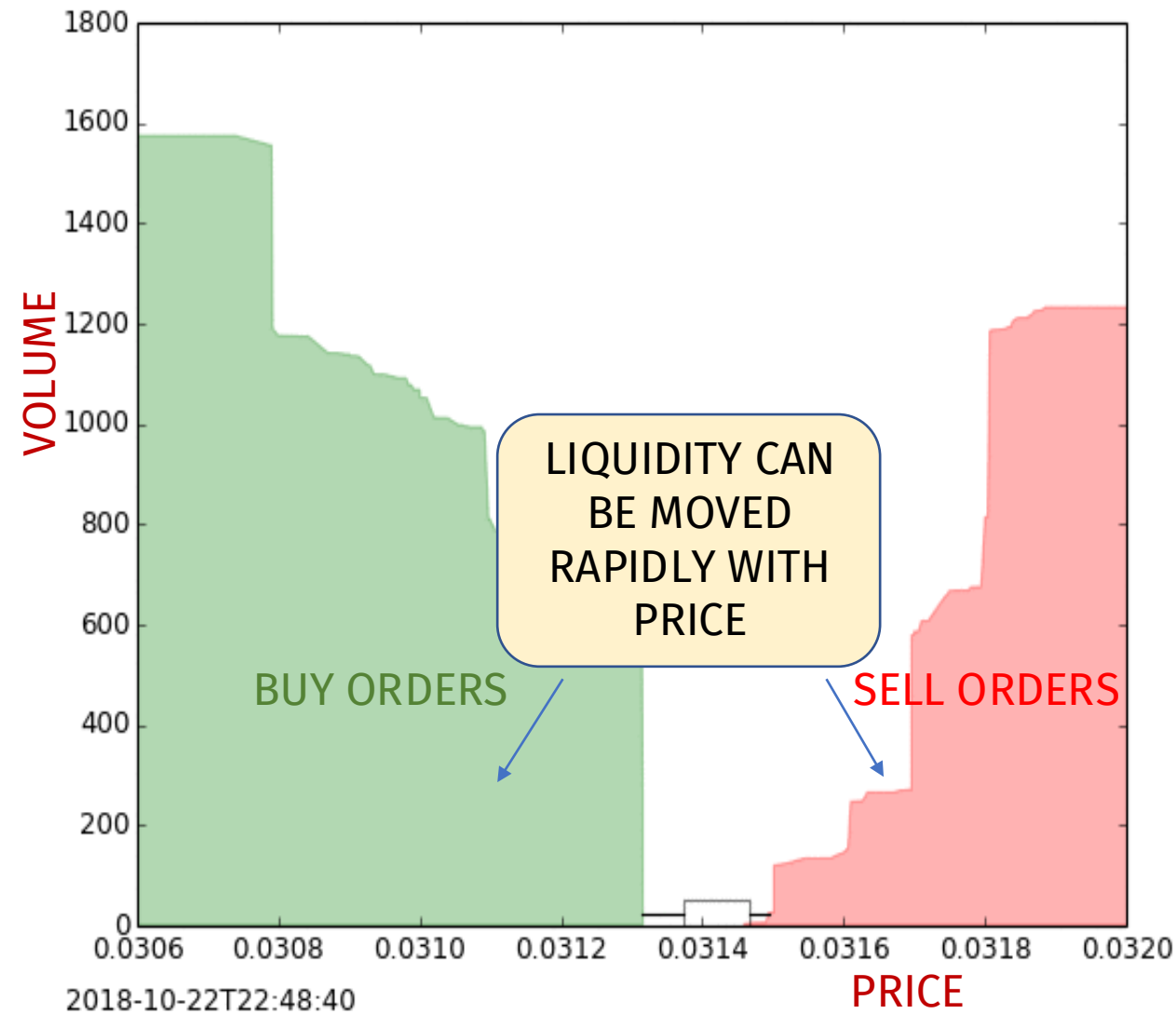
Problems with CFMMs: Capital Inefficiency

- Capital inefficiency : Less capital efficiency than LOBs - why?
- LPs cannot move liquidity around
- Was possible in LOBs



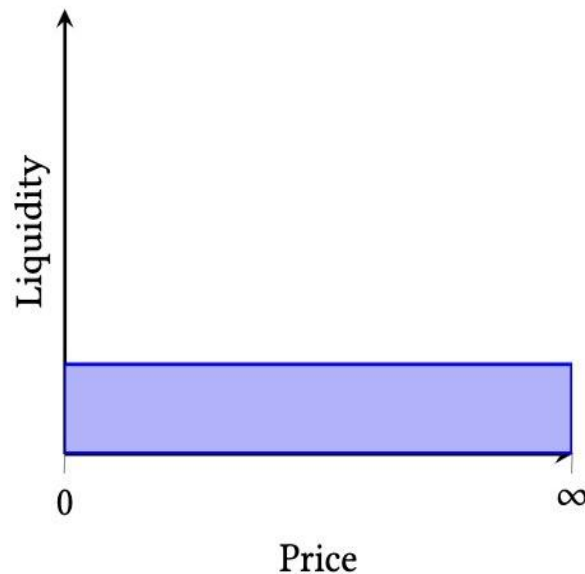
Capital Efficiency in LOBs

- 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

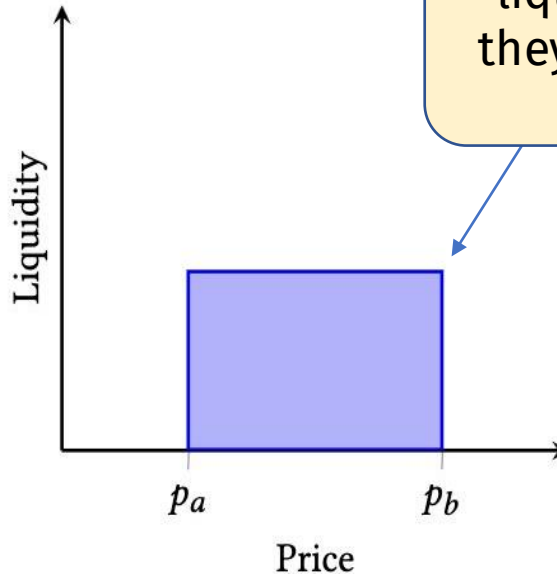


Solution: Concentrated Liquidity

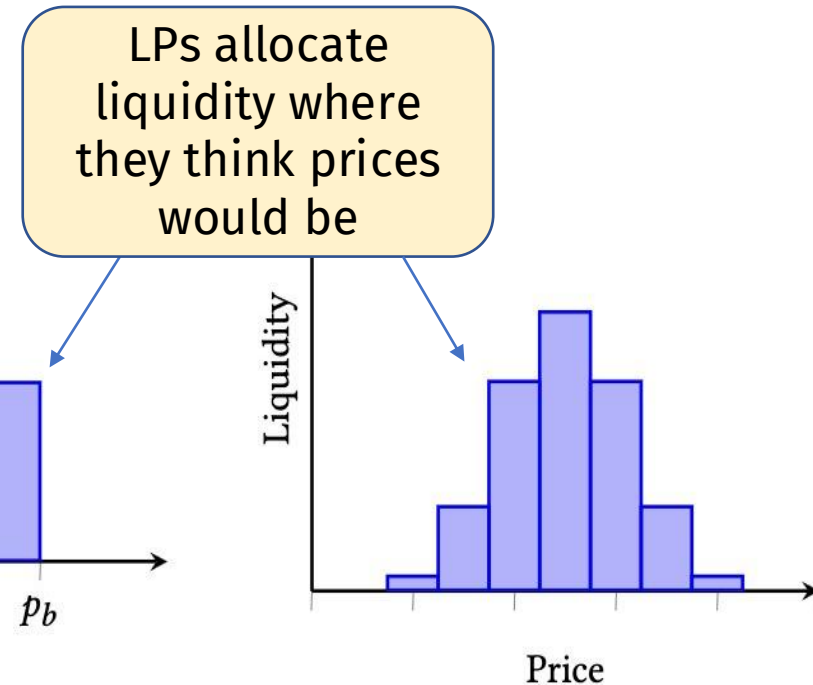
- Allow LPs to specify range of prices
- Divide price range into “buckets”
- LPs choose how much liquidity to allocate to which bucket



(I) UNISWAP v2



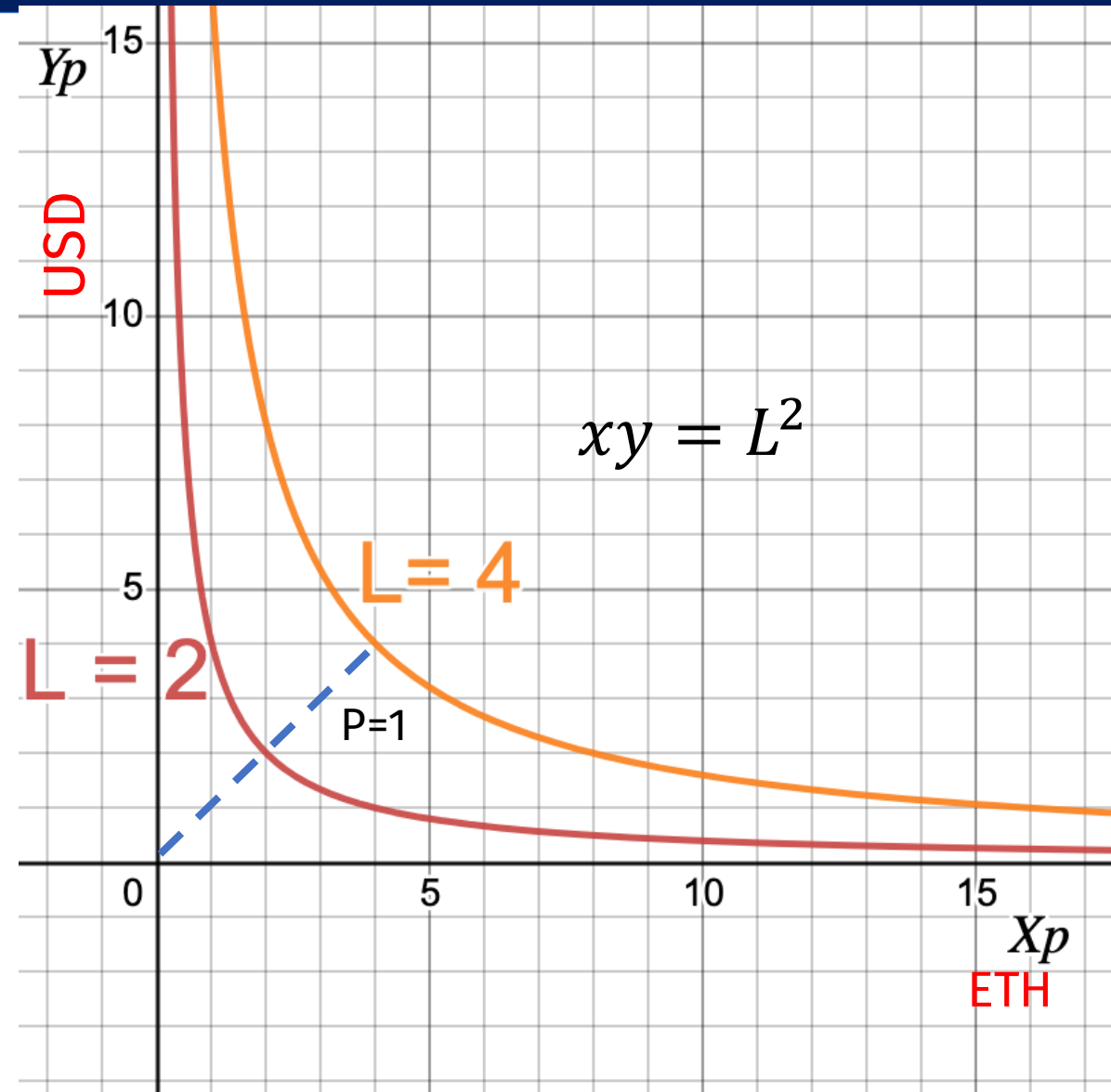
(II) A single position on $[p_a, p_b]$



(III) A collection of custom positions

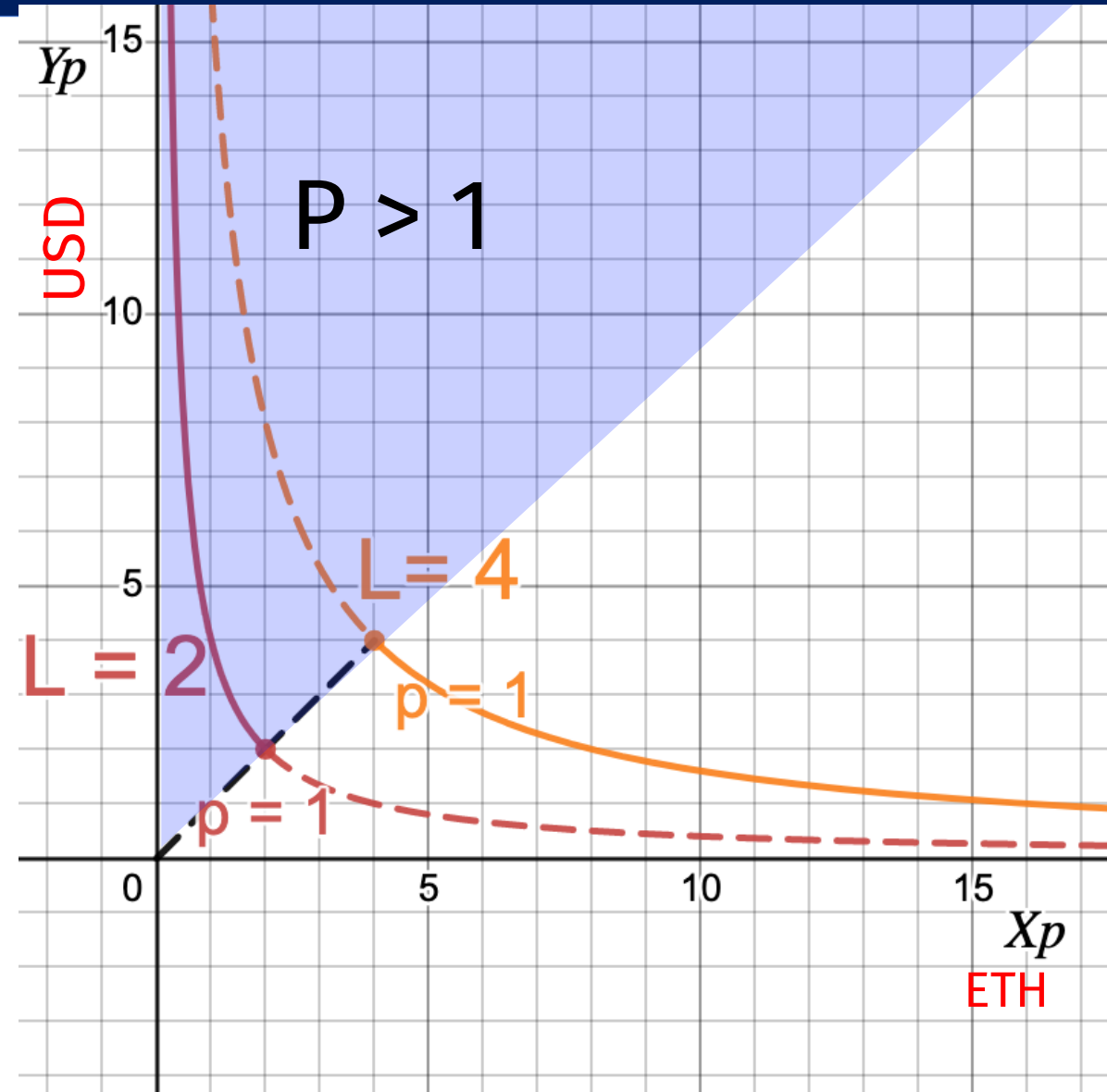
Example

- LP1 wants to add liquidity 2
 - LP2 wants to add liquidity 4
 - How much reserves do they need to add at $p = 1$?
-
- LP1 gives (2 ETH, 2 USD)
 - LP2 gives (4 ETH, 4 USD)
 - The pool has liquidity 6.



Example

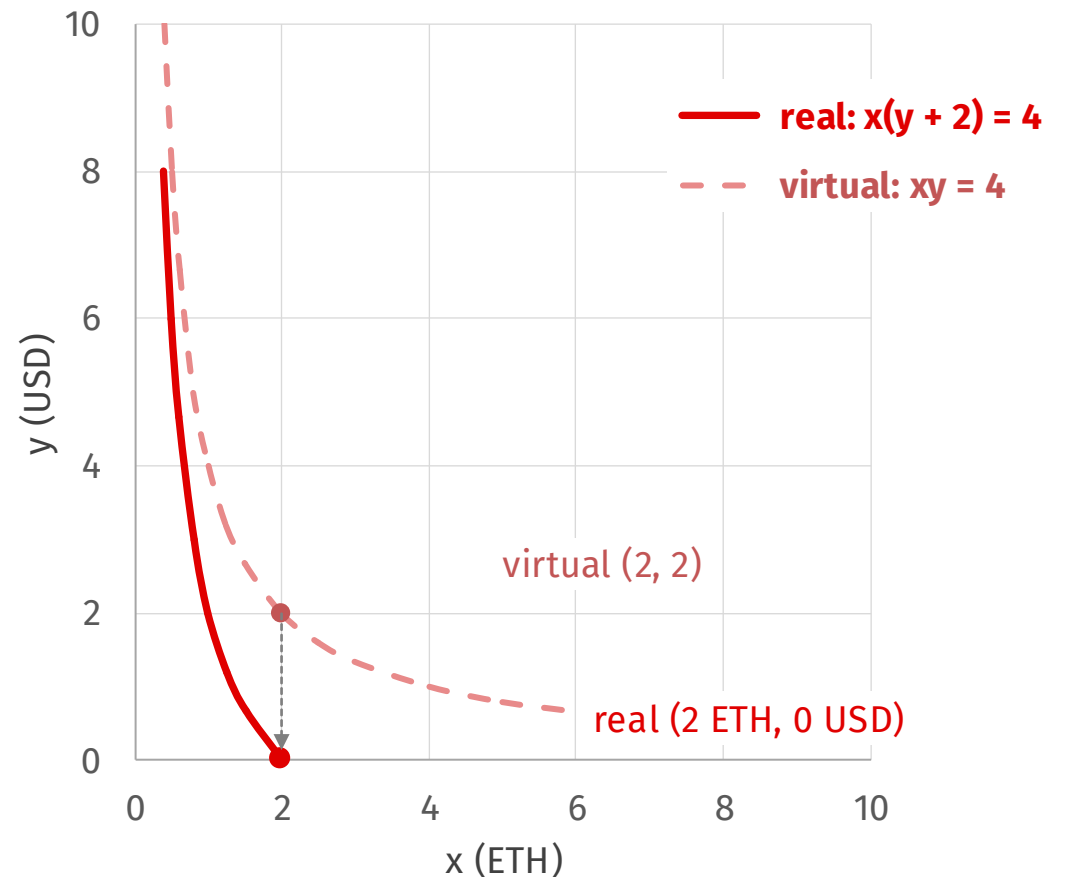
- LP1 adds liquidity 2 for price > 1
- LP2 adds liquidity 4 for price < 1
- What do they need to add?
- LP1 only active when $p > 1$
- Pool gives out ETH in the area!
- LP1 only needs to add:
 - (2 ETH, 0 USD)
- LP2 only needs to add:
 - (0 ETH, 4 USD)



Improving capital efficiency: example

- LP1 only adds: (2 ETH, 0 USD)
- LP1's real reserve follows:
 $(x)(y + 2) = 4$
- I.e., “virtual curve” $xy = L^2$
shifted down by 2
- Why shifting down?
- Note: slope unchanged by shifting.

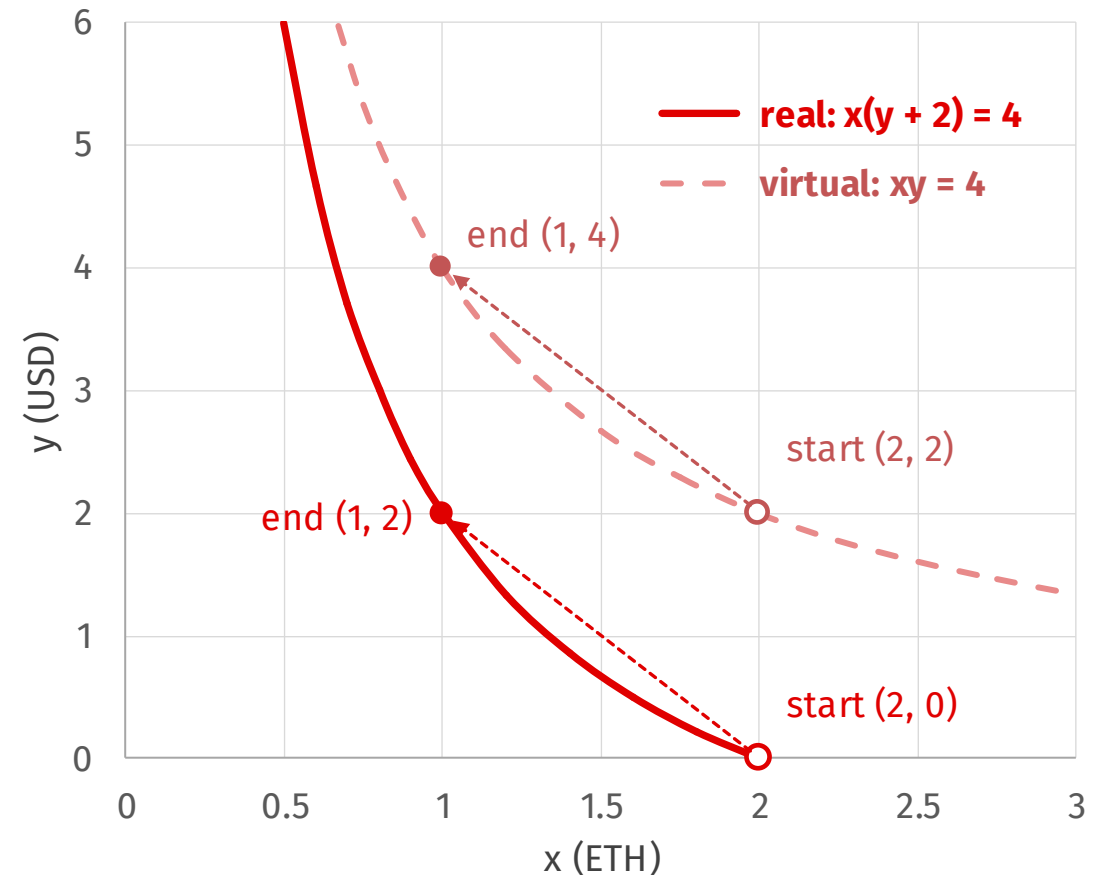
LP1 reserves



Example: buy 1 ETH at $p = 1$

- Buying ETH pushes p up, into LP1's range
- Δy can be computed with both real and virtual curve
 - Real: $(2, 0) \rightarrow (1, 2)$
 - Virtual: $(2, 2)$ to $(1, 4)$
- This example tells us trading on virtual curve is the same as trading on the real curve.

LP1 reserves, before and after the trade

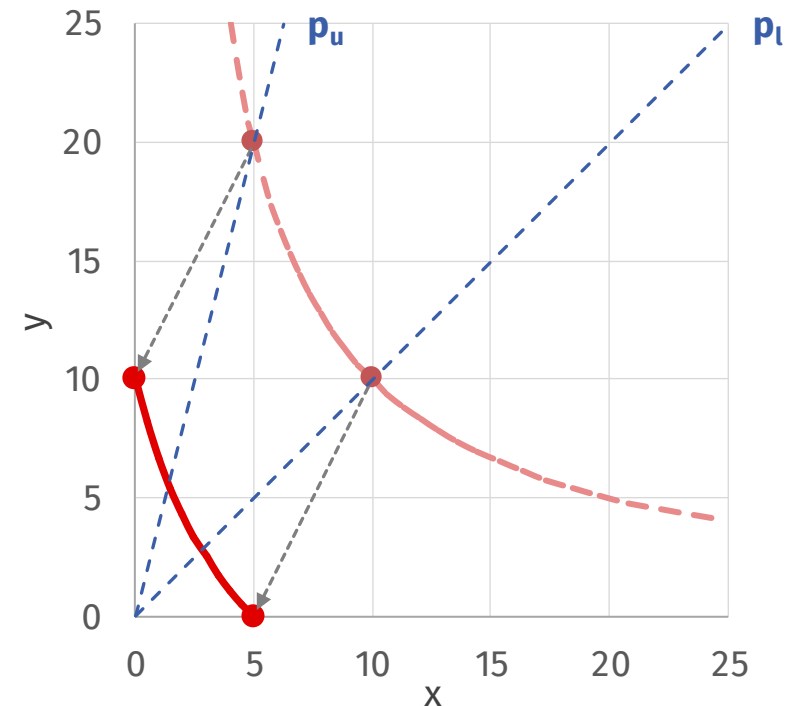


General formula for LP

- In practice, LP specifies assets (x, y) and price range (p_l, p_u)
- From that, we can compute L and the offsets

$$\underbrace{\left(x + \frac{L}{\sqrt{p_u}}\right)}_{\text{Virtual liquidity of ETH}} \underbrace{\left(y + L\sqrt{p_l}\right)}_{\text{Virtual liquidity of USD}} = L^2$$

Price at real reserve (x, y) = price at virtual reserve: $P_x = \frac{y + L\sqrt{p_l}}{x + \frac{L}{\sqrt{p_u}}}$



Exercise

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

- This formula is quite useful
- E.g., What is a position's real reserve (x, y) at price P?
- $x(p) = \frac{L}{\sqrt{p}} - \frac{L}{\sqrt{p_u}}$ // virtual reserve - offset
- Sanity check: holding of X is zero at p_u

LP's Perspective

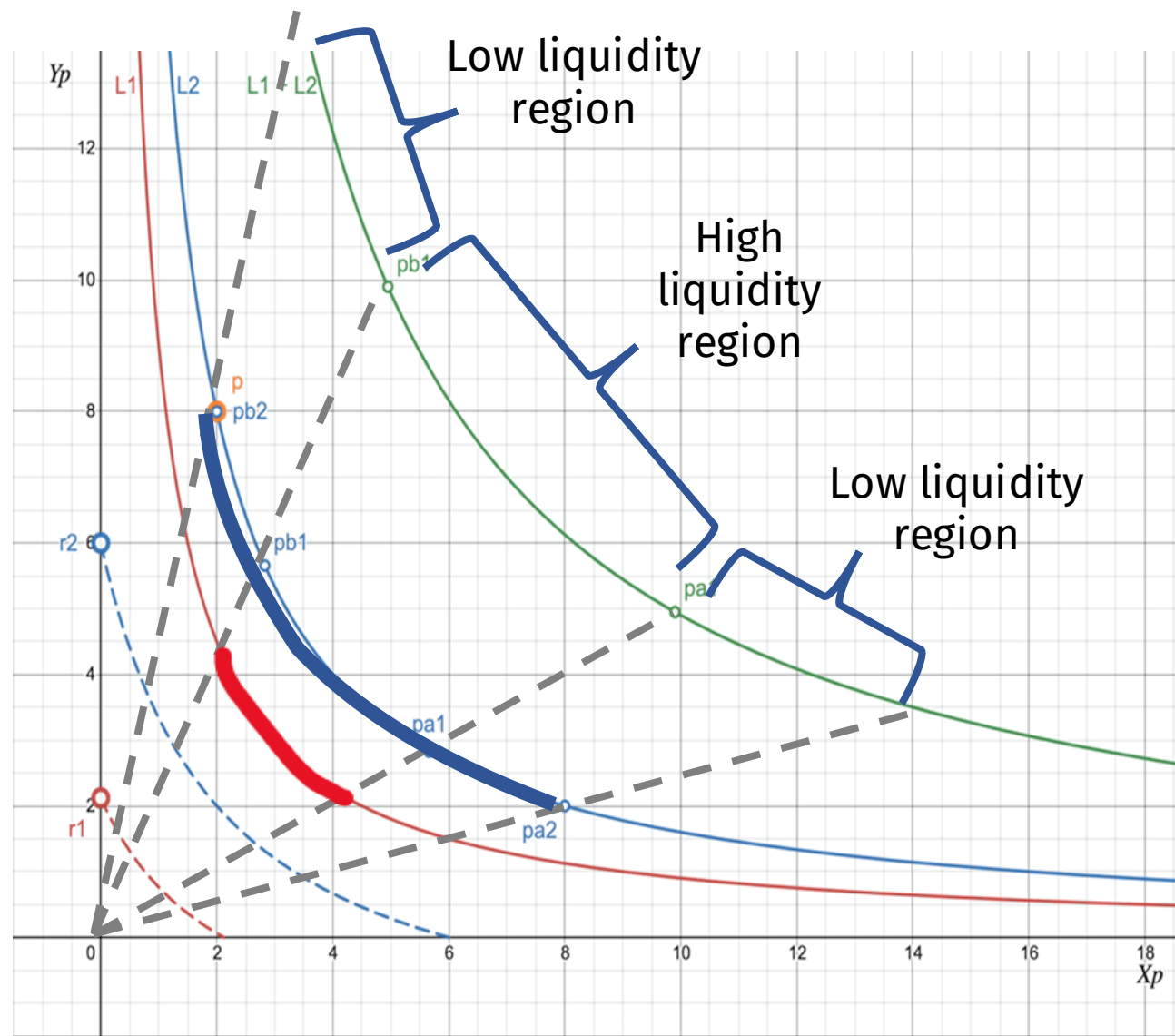
$$\left(x + \frac{L}{\sqrt{p_u}}\right) (y + L\sqrt{p_l}) = L^2$$

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

Trader's Perspective

- 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

Other topics (you can explore in a project)

- **Aggregation** of liquidity across different AMMs for better price for traders
 - DEX aggregators split one trade across pools: 1inch, ParaSwap, Jupiter (Solana)
 - Intent-based solvers compete to fill orders: CoW Swap, UniswapX
- More **flexible curve control** for LPs: e.g., PropAMM
 - PropAMMs on Solana (HumidiFi, SolFi): a market maker reprices the curve every block
 - Uniswap v4 hooks: custom fees and pricing logic per pool
 - Different invariants: Curve StableSwap, Balancer weighted pools

Administrivia

- PS2 out
- No class this Wednesday
 - You can use the time to work on PS2