

Intro

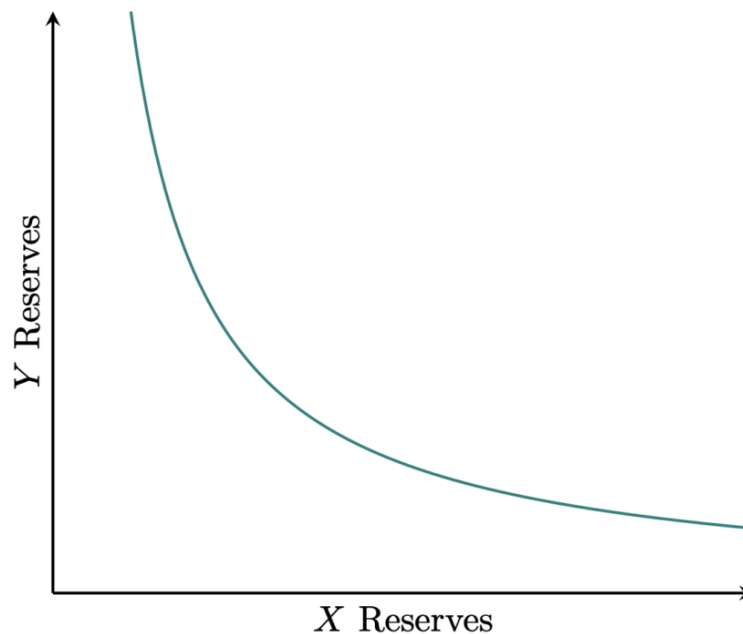
Because V3 introduces the concept of virtual and real liquidities, the setting of positions (adding liquidity) has multiple case scenarios. These scenarios depend on where the current price, **P_c**, is (which will determine the current tick, **i_c**). A tick is a simple marker of price division. For example, when we initiate a pool of X/Y, we can divide the pool in various fractions named ticks. This way the users can add liquidity between the different fractions that correspond to a certain price (in this case we consider the price to be the price of TokenX in terms of TokenY):

$$P(\text{tick}) = 1.0001^{\text{tick}}$$

The relation between tick and pricing is not really important at the moment, what is important is understanding that the price (and as a result the tick) that the contract is currently on changes and the different scenarios that can occur when adding liquidity to a pool, and how these scenarios are related to the current tick/price within the contract.

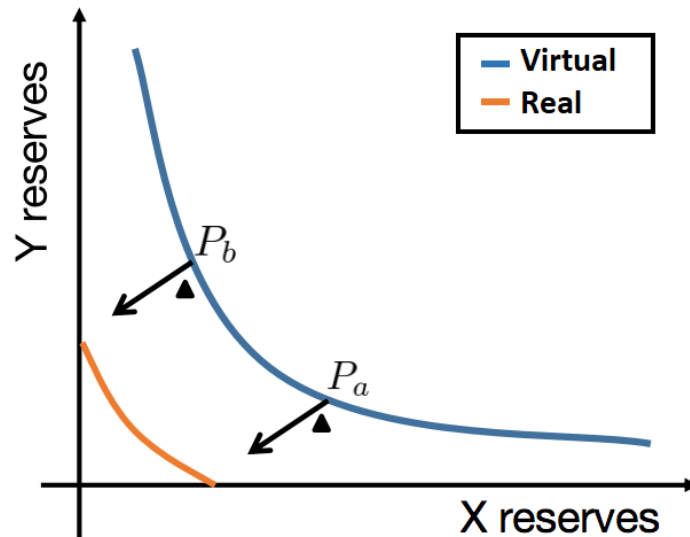
Virtual vs Real liquidity

In uniswap V2, all liquidity is added to the whole pool. For example, if we add to a pool of X/Y, our liquidity will be used if the price of X and Y is between 0 and infinity. This makes it so the liquidity curve always follows $x*y=k$, and that the liquidity is distributed equally between all prices.



However, this is not the case in V3. In V3 you can add liquidity to specific price ranges. For example, I can add liquidity for X prices between 1200 and 1600, or for Y between 10k and 50k.

So in practice, between each price range, we will have different k to satisfy $x*y=k$. This makes it so the liquidity curve is not uniform. The whole pool is then represented with a virtual liquidity curve, but the real curve used is the one in the price range we are currently in.



Scenarios

So, when we are setting a new position, we can have 3 different situations depending on where the price/tick of the pool is currently at, **P_c and i_c**, and what are the upper and lower prices, **P_u and P_l** (and ticks, **i_u and i_l**) in which we want to set a new position. This is shown in this highlighted section of the Uniswap V3 whitepaper:

$$\Delta Y = \begin{cases} 0 & i_c < i_l \\ \Delta L \cdot (\sqrt{P} - \sqrt{p(i_l)}) & i_l \leq i_c < i_u \\ \Delta L \cdot (\sqrt{p(i_u)} - \sqrt{p(i_l)}) & i_c \geq i_u \end{cases}$$

$$\Delta X = \begin{cases} \Delta L \cdot \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right) & i_c < i_l \\ \Delta L \cdot \left(\frac{1}{\sqrt{P}} - \frac{1}{\sqrt{p(i_u)}} \right) & i_l \leq i_c < i_u \\ 0 & i_c \geq i_u \end{cases}$$

The unusual scenarios ($i_c < i_l$ or $i_c \geq i_u$) are harder to understand because they are incomprehensible from a V2/normal AMM perspective. To help understand these cases, we can review two very important parts of the whitepaper. Lets call them Rule 1 and 2 for further reference:

$$x \cdot y = k \quad (2.1)$$

Specifically, a position only needs to hold enough of asset X to cover price movement to its upper bound, because upwards price movement¹ corresponds to depletion of the X reserves. Similarly, it only needs to hold enough of asset Y to cover price movement to its lower bound. Fig. 1 depicts this relationship for a position on a range $[p_a, p_b]$ and a current price $p_c \in [p_a, p_b]$. x_{real} and y_{real} denote the position's real reserves.

When the price exits a position's range, the position's liquidity is no longer active, and no longer earns fees. At that point, its liquidity is composed entirely of a single asset, because the reserves of the other asset must have been entirely depleted. If the price ever reenters the range, the liquidity becomes active again.

The amount of liquidity provided can be measured by the value L , which is equal to \sqrt{k} . The real reserves of a position are described by the curve:

$$\left(x + \frac{L}{\sqrt{p_b}}\right)(y + L\sqrt{p_a}) = L^2 \quad (2.2)$$

This curve is a translation of formula 2.1 such that the position is solvent exactly within its range.

Rule 1

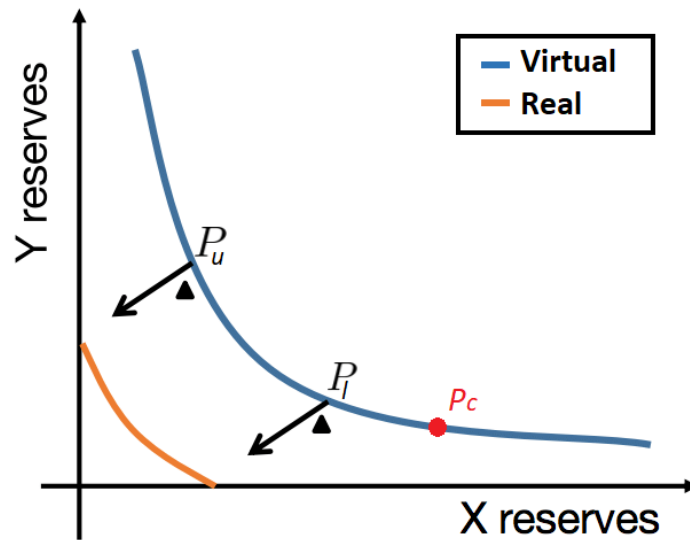
And add an also very important part provided at the end of the whitepaper:

The amount of token0 (ΔX) or token1 (ΔY) that needs to be deposited can be thought of as the amount that would be sold from the position if the price were to move from the current price (P) to the upper tick or lower tick (for token0 or token1, respectively).

Rule 2

While reading the scenarios, please refer back to these rules, and keep in mind equation 2.2. All of this will be extremely relevant.

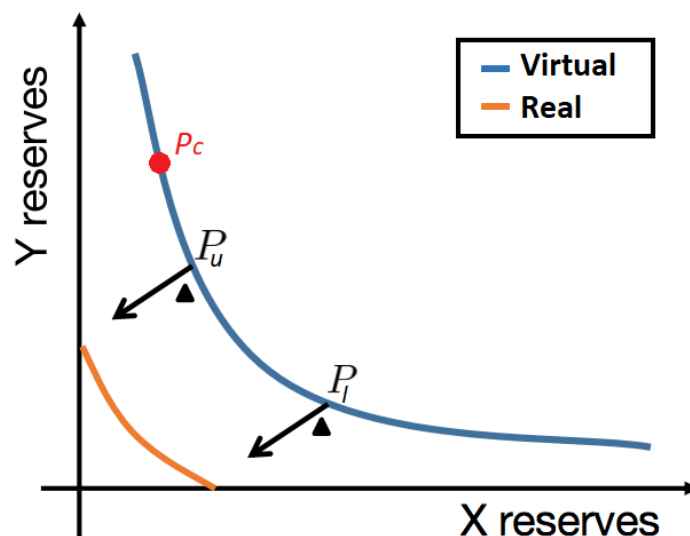
Scenario 1: Current price(and tick) is under the lower tick of the new position being set ($i_c < i_l$)



In this case, we can see that the real reserves of token Y are zero for the current price of TokenX relative to TokenY (P_c).

So if we revisit rule 2 and focus on TokenX. The rule tells us that the amount to be deposited is the amount of tokens that would need to be sold if we want to move from the current price, P_c , to the upper price, P_u . Since the position has no Y tokens in the current price, no Y tokens would be sold, and so no Y tokens will be deposited in this case.

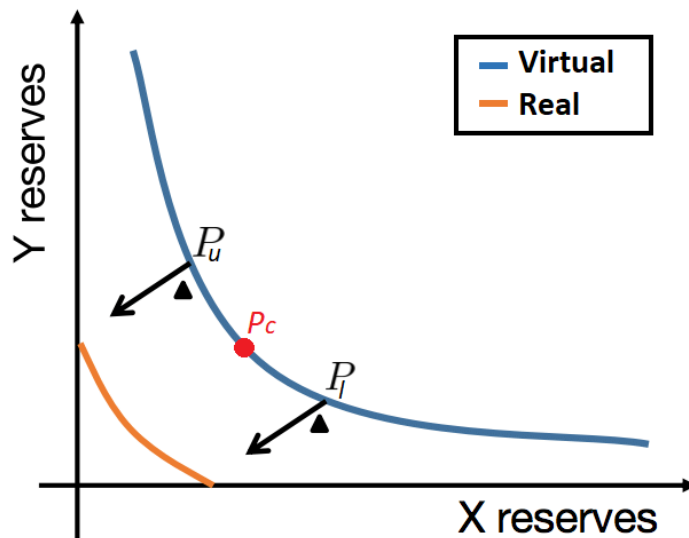
Scenario 2: Current price(and tick) is over the upper tick of the new position being set ($i_c > i_u$)



In this case, we can see that the real reserves of token X are zero for the current price of TokenX relative to TokenY (P_c).

If we again apply the thinking of Rule 2, then the amount of tokenX to be sold would be zero because there are none in the current position, so we will deposit 0 x tokens.

Scenario 3: Current price(and tick) is between the upper and lower ticks of the new position being set



In this case, setting a position will require a transfer of both Y and X tokens.

Conclusion

If one of the tokens is not being transferred, it simply means that we need to add liquidity to other positions in the curve, according to the rules previously specified.

The current tick the contract is in, can be seen in its storage under the name **cur_tick_index**. Simply check it and add liquidity as needed (see the first image in the Scenarios section).