Uniswap V2 Pricing

March 14, 2022

In this paper, I will dive into the constant product formula that defines Uniswap V2. I will then discuss how tokens are priced on Uniswap V2 and incorporate fees as well. Finally, I will go over why high amounts of liquidity are important for liqudity pools and its relation to slippage.

1 Preliminaries

Uniswap V2 is an automated market-maker platform (AMM) where smart contracts handle transactions between two parties; while the infrastructure behind Uniswap V2 was crucial for its success in the Decentralized Finance (DeFi) space, I will focus on the price mathematics that made such an AMM possible.

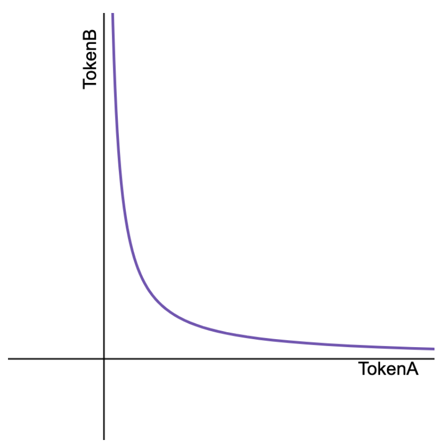

Figure 1: Arbitrary Graph of a Uniswap V2 Liquidity Pool

All liquidity pools on Uniswap V2 consist of two tokens TokenA and TokenB which are stored in the `UniswapV2Pool.sol` smart contract associated with the liquidity pool. When a liquidity pool is first initialized, its `k` value is determined by the product of the amount of TokenA (`reserve0`) and the amount of TokenB (`reserve1`) deposited. This `k` value increases throughout the lifetime of the smart contract as a result of Uniswap V2's 0.3% trading fee.

`k` is a point on the graph in Figure 1 and as a result, regardless of the trades that the smart contract executes, `k` must always remain on the graph.

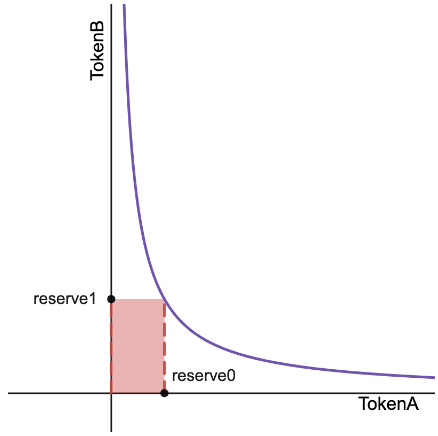

Figure 2: Calculating k value

Figure 2 above makes it evident that using basic geometry, we can calculate the price of k using the following equation:

$$reserve0 \cdot reserve1 = k$$

Uniswap's official documentation [1] utilizes the equation $x \cdot y = k$, where $x = reserve0$ and $y = reserve1$; for the rest of this paper we shall refer to Uniswap's convention.

We now delve into how pricing is determined on Uniswap V2. Assume that I hold Δx of token A, where $\Delta x > 0$ and I want to exchange my Δx for an amount of token y defined as Δy. The question is then, how much Δy can I receive for my Δx?
We return to our equation $x \cdot y = k$; if we are to incorporate the changes in `reserve0` and `reserve1` into our equation, we get the following:

$$(x + \Delta x)(y - \Delta y) = k$$

Utilizing algebra, we can derive the following formula:

$$\Delta y = y - \frac{k}{x+\Delta x} \leftarrow (1)$$

This is the foundation of all AMMs: using mathematical formulas, one can determine the return of an exchange without the need of a central authority. Uniswap V2's constant product formula is a example of such a mathematical formula but other DEXs such as Curve utilize different formulas to best serve their needs. Regardless of the equation utilized, a question arises about the consequences of when the return value of an exchange is not equal to Δy. In this scenario, if I trade Δx of Token A for Token B, let ω represent my real return while Δy is the theoretical output using (1). We can examine this with the following two cases:

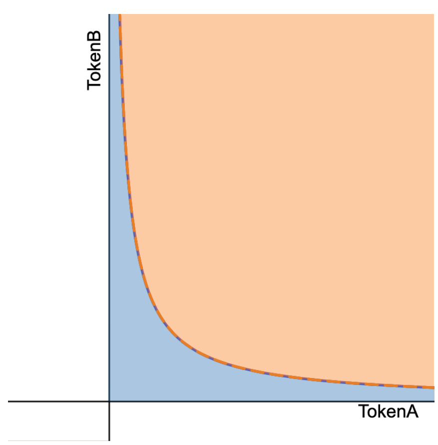

Figure 3: Scenario of $\Delta y \neq \omega$

Case 1 Assume $\omega < \Delta y$.
Let $k = (x + \Delta x)(y - \Delta y)$ and $k' = (x + \Delta x)(y - \omega)$. Under this assumption, we can state the following inequality:

$$(x + \Delta x)(y - \Delta y) < (x + \Delta x)(y - \omega) \leftarrow \text{by fact that } (y - \Delta y) < (y - \omega)$$
$$k < k'.$$

Because $(x + \Delta x)$ is constant, this means that the y-coordinate of k' will be above the graph.

In this scenario, $\omega < \Delta y$ implies that the trader losing out on this trade. Therefore, any hypothetical k point above the graph (orange area of Figure 3) implies that the smart contract is the winner in this trade.

Case 2 Assume $\omega > \Delta y$. Let $k = (x + \Delta x)(y - \Delta y)$ and $k' = (x + \Delta x)(y - \omega)$. Under this assumption, we can state the following inequality:

$$(x + \Delta x)(y - \Delta y) > (x + \Delta x)(y - \omega) \leftarrow \text{by fact that } (y - \Delta y) > (y - \omega)$$
$$k > k'.$$

Because $(x + \Delta x)$ is constant, this means that the y-coordinate of k' will be below the graph.

In this scenario, $\omega > \Delta y$ implies that the smart contract is losing out on this trade. Therefore, any hypothetical k point below the graph (blue area of Figure 3) implies that the trader is the winner in this trade.

With all prerequisites being explained, we now dive into how pricing and fees work on Uniswap V2.

2 Pricing and Fees

2.1 Reserve Ratio

As described above, the constant product formula is the main method of how pricing is determined on Uniswap V2. However, another method that traders may utilize is pricing via calculating the ratio of the reserves. To introduce this section, I shall utilize a similar concept from microeconomics:

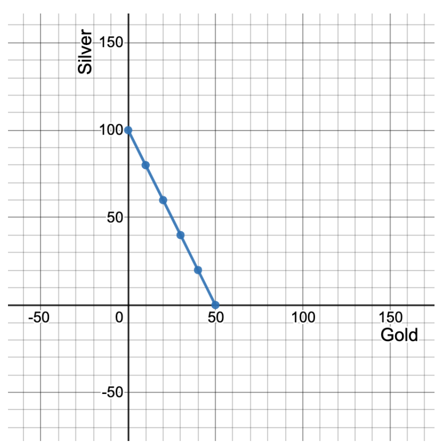

Figure 4: Production Possibilities Curve of Gold and Silver

Assume that an alchemist is able to produce both gold and silver, but is limited by the amount of time he has. In an hour, he can make either 100 bars of silver or 50 bars of gold. From figure 4, we can further deduce that for every bar of gold he makes he gives up the possibility to make 2 bars of silver and vice versa, for every bar of silver he makes he gives up the possibility to make 0.5 bars of gold. Therefore, using the slope of the production possibilities curve, we can derive the following:

$$\text{opportunity cost of gold} = \frac{2}{1}$$
$$\text{opportunity cost of silver} = \frac{1}{2}$$

Applying this idea over to Uniswap V2, it is evident that tokens in a liquidity pool can be priced in the same manner. If a liquidity pool ab contains tokens a (`reserve0`) and b (`reserve1`), then the price of these tokens are as follows:

$$\text{price of token } A = \frac{\text{reserve1}}{\text{reserve0}}$$
$$\text{price of token } B = \frac{\text{reserve0}}{\text{reserve1}}$$

For those wanting to calculate liquidity pools prices for applications such as directed graphs, this is more convenient than using the constant price formula. However, this pricing is prone to change every time a trade occurs and as a result, one must update these ratios.

2.2 Trading Fee

The constant product formula described in Preliminaries, while simple, did not account for Uniswap V2's 0.3% trading fee. The majority, if not all, of this trading fee goes to compensating liquidity providers for providing liquidity to the pool. This trading fee is applied to the input token of a trade; therefore, the formula for calculating Δy is the following:

$$\Delta y = y - \frac{k}{x+(0.997 \cdot \Delta x)}$$

If we are trading Δx of token a for some token b, then 99.7% of our quantity of token a is actually utilized; the other 0.3% is deposited in `reserve0`. This is the reason why k increases after every trade and why the graph of k shifts after every trade: our Δy is proportional to $0.997 \cdot \Delta x$ rather than Δx

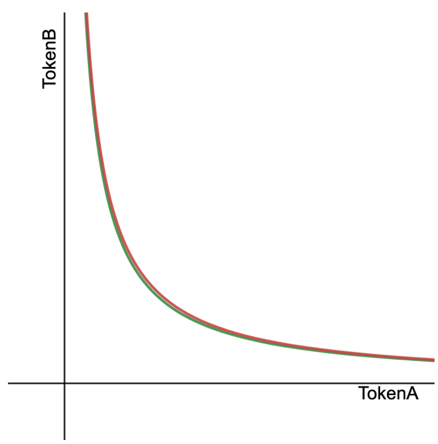

Figure 5: Hypothetical Graph of k before a trade (colored in green) and after a trade (colored in red)

3 Liquidity and Slippage

A trader who wants to receive a fair bargain in a trade is incentivized to trade in liquidity pools with high amounts of liquidity. To illustrate why, listed below are two hypothetical cases that emphasize why liquidity is important:

Case 1 Let Alice and Bob be two Uniswap V2 traders and let Alice engage in a trade on liquidity pool ab prior to Bob. Listed below are the statistics of liquidity pool ab:

- `reserve0` $= 4$
- `reserve1` $= 6$
- $k = 24$

If Alice wants to trade 1 token of token a, then we can calculate her return:

$$\Delta y = 6 - \frac{24}{4+1} = 1.2$$

Now assume Bob engages in a trade on liquidity pool ab with 1 token of token a as well. Listed below are the statistics of liquidity pool ab, accounting for Alice's trade:

- `reserve0` $= 5$
- `reserve1` $= 4.8$

- $k = 24$

We now calculate Bob's return:

$$\Delta y = 4.8 - \frac{24}{5+1} = 0.8$$

If Alice and Bob both expected a return of 1.2, then Alice will get a fair trade here while Bob will experience slippage and receive a worse trade relative to Alice.

Case 2 Let Alice and Bob be two Uniswap V2 traders and let Alice engage in a trade on liquidity pool cd prior to Bob. Listed below are the statistics of liquidity pool cd:

- `reserve0` $= 40000$
- `reserve1` $= 60000$
- $k = 2.4 * 10^9$

If Alice wants to trade 1 token of token c, then we can calculate her return:

$$\Delta y = 60000 - \frac{2.4*10^9}{40000+1} = 1.49996$$

Now assume Bob engages in a trade on liquidity pool cd with 1 token of token c. Listed below are the statistics of liquidity pool cd, accounting for Alice's trade:

- `reserve0` $= 40001$
- `reserve1` $= 59998.5004$
- $k = 2.4 * 10^9$

We now calculate Bob's return:

$$\Delta y = 59998.5004 - \frac{2.4*10^9}{40001+1} = 1.49989$$

In this scenario, if both Alice and Bob are expecting a return of 1.49996, then Alice will get a fair trade while Bob will experience an extremely small amount of slippage and will receive a slightly worse trade relative to Alice.

Utilizing the concept of percent difference, we see percent difference of Alice's and Bob's return in case 1 is 40% while in case 1, the percent difference is 0.0046669%. Such examples emphasize the importance of liquidity as a a liquidity pool with low amounts of liquidity is bound to expose traders to high amounts of slippage. A liquidity pool with high amounts of liquidity, on the contrary, is much less susceptible to high price fluctuations and as a result, traders who are exposed to slippage will be exposed to a small amount of it.

4 Conclusion

Uniswap V2 has been embraced by the DeFi community and is still going strong almost two years after its release. However, such a DEX does not come without its flaws. The most particular flaw of Uniswap V2 is its inefficient use of capital; since liquidity is spread evenly across the graph of k [2], this implies that for certain liquidity pools whose price does not fluctuate by much, capital is left sitting. In a liquidity pool such as `USDC/USDT`, most exchanges will occur at the $1 price range and therefore, tokens at other price ranges will almost never be utilized. However, this issue is clearly addressed with the introduction of Uniswap V3, which introduces the concept of concentrated liquidity to better utilize capital. Regardless of such flaws though, Uniswap V2 certainly has left its mark not just on decentralized `ERC-20` to `ERC-20` transactions on the Ethereum blockchain, but on DeFi based on other blockchains as well.

References

[1] https://docs.uniswap.org/protocol/V2/concepts/protocol-overview/how-uniswap-works. [Online; accessed 10-March-2022].

[2] https://mvpworkshop.co/blog/uniswap-v3-explained-all-you-need-to-know/#Uniswap_v3_vs_v2_-_Efficiency_Security_and_NFTs. [Online; accessed 11-March-2022].