1 Intro

In this guide, I'll begin with how order books on centralized exchanges actually work, and how automated market makers (AMMs) allow us to cut out the centralized middlemen and their order books, and trade peer-to-peer. Then I'll go into liquidity pools, which is the core innovation that makes AMMs possible. Finally, I will try to clearly explain **impermanent loss**, which is critical to understanding and assessing risk as a liquidity provider, and which is often misunderstood.

Feel free to skip ahead to whatever sections interest you. It starts with the basics, but gets well into the weeds by the end.

2 Order books

Centralized exchanges like Binance or Coinbase or the NYSE use **order books** to facilitate transactions between buyers and sellers. Order books are essentially lists of limit buy orders and limit sell orders that clients have placed. A limit order is an open offer to buy or sell some amount of an asset at a price specified by the customer placing the limit order.

These limit order get consumed by market orders, which is the instant kind of order that happens when you just click "Buy" or "Sell". If you execute a market buy, you will immediately buy from whatever limit sell order in the book currently is offering the lowest price. If you execute a market sell, you will immediately sell to whatever limit buy order in the book currently is offering the highest price. These market orders consume part or all of the limit order that was offering the best deal.

The exchange profits by charging fees in return for the service of matching these market orders with these sell orders.

The official price of the asset at any time is simply the cheapest limit sell order on the books at the time (ie: the current best bargain, or how cheaply you could buy some of this asset *right now*).
The most expensive limit buy order on the books is always lower-priced than the cheapest limit sell order, and the difference between them is known as the spread.
Price moves when market buy orders entirely consume the cheapest limit sell order. When this happens, the price becomes whatever price is offered by the *next* cheapest limit sell order.

For example, imagine if the cheapest limit sell order currently on the books (ie: the best bargain, which is what defines the current price) is a whale offering to sell 100 BTC at $40k.

Now imagine that another whale makes a market buy for 70 BTC. They would get all 70 at the price of 40k, and this actually wouldn't change the price by one penny, because 30 BTC are still being offered by the first whale at $40k, so $40k is still the best bargain, and thus is the price.

However, if instead the buyer bought 140 BTC, they would fully consume the seller's limit sell order, and the price would now become whatever the next-cheapest limit sell order is priced at (for example, $40,000.50), and the remaining 40 BTC that the buyer buys will thus be bought at a higher price than what the first 100 BTC were bought for (when the price changes in the middle of a transaction like this, it is called **price slippage**).

It is even possible that when the cheapest limit sell order is consumed, the next cheapest limit sell order is at a significantly higher price, **causing the price to instantly teleport a great distance**. This generally happens when there is low liquidity, which means a low density of limit orders on the books. This is why high **liquidity is important for a healthy market**.

This is an important concept to keep in mind: the price of a stock is NOT a weighted average of the price to which all the available stocks have been bought during time, but reflects **only the last price** to which stocks have been exchanged. The '**market capitalization**' is for sure an important factor when considering an investment, and it's the actual price of a stock multiplied for the total number of available stocks on the market. The crazy thing in my opinion, is to suppose that, especially if many stocks are available, the price of ALL stocks is that one ... should everyone try to sell its stocks at that specific price, would they all get sold? If there is hypothetically a dead market moment, and a single guy sells (to himself, having two different accounts) 1/1.000.000 of a Bitcoin for 1.000.000$ per Bitcoin, is the price for a Bitcoin now 1 million? Apparently yes, that's exactly how it works. So an extremely low exchange market, can be subjects to such problems and will be MUCH more volatile.

This concludes how order books work on centralized exchanges. As you can imagine, since decentralization is a major theme in crypto, there has long been a desire to find a peer-to-peer alternative. This was finally made possible with the invention of the **automated market maker**, which led to the birth of DeFi. **AMMs** were inspired by the structure of traditional stock dealer markets like the Nasdaq (rather than broker markets like the NYSE).

3 Automatic market makers and liquidity pools

AMMs are the innovation that lies at the core of every decentralized exchange, like UniSwap, SushiSwap, PancakeSwap, Curve and hundreds of others. AMMs use **smart contracts** to create an automatic, decentralized, peer-to-peer alternative to order books, allowing people to trade assets without going through CEXes.

The central idea of AMMs is a concept called liquidity pools. Each liquidity pool in an AMM allows people to trade a specific asset pair (like ETH/USDC) in either direction. In other words, an ETH/USDC liquidity pool would allow you to buy ETH with USDC or buy USDC with ETH. AMMs are made up of large amounts of these liquidity pools, allowing for large amounts of possible trade pairs.

Each liquidity pool is made up of equal portions of the trading pair's two assets. These pools are filled by liquidity providers, who are people like you and me who choose to supply their assets to facilitate trades by other people, in order to earn rewards in the form of trading fees.

When a trader uses the pool to make a swap, they are really just adding some amount to one of the two assets in the pool, and taking out the corresponding amount of the other asset in the pool. The trader also pays a trading fee, which is what rewards all the liquidity miners in that pool (they share the fee, weighted in proportion to how much of the pool each provider is providing).

As a side note, liquidity providers also sometimes get rewarded in a separate way if they provide liquidity to "incentivized pools". Sometimes, when some DEX or DeFi ecosystem is new, they will temporarily offer incentives to liquidity providers out of their own pocket in order to attract traders and gain a larger slice of the DeFi world, to profit more in the long run. These incentives usually follow a diminishing returns type of curve. Getting these rewards is called liquidity mining, and it is the central strategy in yield farming.

The description of liquidity pools I have provided so far is something a lot of you will have heard before. But it is missing a few key mechanics that I think are important to understand. If you are very sharp, then you might have thought of one or two questions when reading my explanation so far.

The two questions that I think we need to get to the bottom of before we truly understand liquidity pools are: what happens when the two halves of the pool are put out of balance due to traders using the pools to swap, and how does the pool know what relative price to use between the two assets ?

These are highly related questions. Here is the key: no matter what, the pool itself *always* considers the two sides of the pool (for example, the ETH side and the USDC side) to be of equal value. The following example is **EXTREMELY important** to better understand the concepts that usually lie behind AMM algorithms.

So, let's say you decide to buy ETH with USDC using a DEX. You want to spend $4000 USDC. The amount of ETH that will get you will depend on the ratio between the amount of ETH and the amount of USDC in the pool, and nothing else. Let's say the pool currently contains 1,000,000 USDC and 500 ETH. That is a ratio of 2000 USDC per 1 ETH. That means, in the pool's opinion, the price of ETH in USDC is 2000, regardless of what the outer world of CEXes and other DEXes might believe.

So, after your trade, you end up with 2 ETH, and the pool now contains 1,004,000 USDC and 498 ETH (plus a tiny bit extra, because your trading fee actually just gets added to the pool, and the providers will get their share of it whenever they pull their liquidity out).

Now the ratio of USDC to ETH in the pool is 2016, so the price of ETH in the pool's opinion is now 2016 USDC, and the price of USDC in the pool's opinion is 0.000496 ETH.

This brings us to a very key concept. The price of ETH in the pool's opinion has gone up to 2016 due to your trade, **but this price spike didn't happen in the rest of the world of CEXes and DEXes**! Therefore, the rest of the world probably still agrees that ETH costs about 2000 USDC, which brings an arbitrage opportunity: people can now buy discount USDC with their ETH from the pool in our example, and then use it to buy back their ETH plus a little extra on any other exchange. When people take advantage of this arbitrage opportunity, it pushes the price of ETH down (or equivalently the price of USDC up) in the eyes of the pool, reversing the effect of your trade, because they are adding ETH and removing USDC from the pool, bringing the ratio back towards 2000: 1.

The following two facts are extremely key:

1. The prices of the two assets in a pool are determined entirely **by the ratio between their amounts**. For example, if our pool somehow ended up containing 1 ETH and 1 million USDC (wouldn't happen because people would take advantage of arbitrage long before we could get there), then the price of ETH in that pool would be 1 million USDC, regardless of the rest of the world.
2. These arbitrage trades are the one and only thing that serve to rebalance the ratios of pools to keep the prices on DEXes more or less in lockstep with all other DEXes and CEXes. It basically makes it so that the average price in the eyes of the entire world acts as a point of gravity for any specific pool.
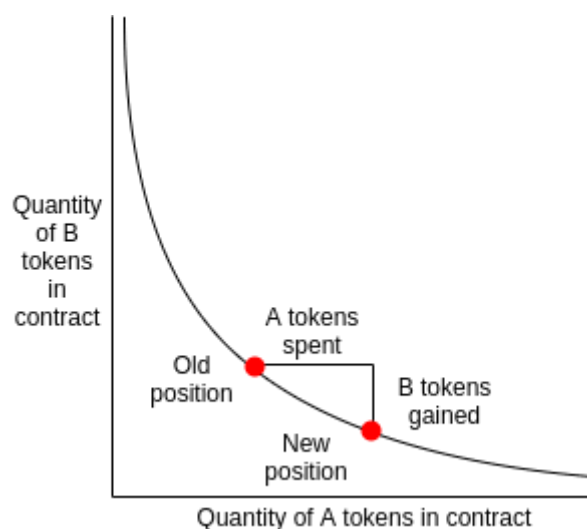
4 Being a liquidity provider

Generally speaking, anyone can create a new liquidity pool to allow others to trade some specific pair. Once a pool has been made, anybody can provide liquidity to it, or withdraw their liquidity, at any time. When you provide liquidity, you must provide the two assets in equivalent amounts (at least, in the eyes of the pool, determined by the current ratio of the pool).

When you provide liquidity, the funds leave your wallet, unlike with staking. This is necessary, because these funds need to be mobile to facilitate swaps.

So, how does the pool know that some portion of its liquidity belongs to you?

When you add liquidity to a pool, it will give you some amount of a special token called an LP (**Liquidity Pool**) token. The token will be specific to the asset pair, and will be called something like **LP-ETHUSDC**. They will also be **specific to the AMM** you are using.

These LP tokens are managed in such a way that the amount of this token that you, a liquidity provider, hold, is proportional to your slice of the pool. In other words, if you are providing 10% of all the liquidity in a pool, you will also have 10% of all LP-ETHUSDC tokens that exist on that AMM.

When you want to cash out, you trade in your LP tokens, and that lets the pool know how much ETH and USDC to give you back (in this example, you would get 10% of the ETH and 10% of the USDC in the pool, because you traded in 10% of all existing LP-ETHUSDC tokens, proving you owned 10% of the pool). Note that **trading fees are always just added to the pool**, making the total holdings of the pool go up, which means that when a liquidity provider pulls out their liquidity, the fees they earned while they were providing liquidity are naturally part of the share of the pool they have claim to.

Note that when you add your funds to a liquidity pool, you are taking on risk that the smart contract of the specific AMM you are using can be exploited. You are also exposed to a change in price of the two assets you are providing, because when you pull out your liquidity, it is given back to you in the form of those two assets. So it's like you were holding them all along.

So, in our example above, we are exposed to ETH price movements, we are exposed to USDC permanently losing its peg, and we are exposed to vulnerabilities in the smart contract of the DMM.

We are also always exposed to **one more key risk**.

5   Impermanent Loss

Impermanent loss is a way that you can lose money when providing liquidity. More accurately, it refers to losing money *relative to if you had just held the two assets you provided to the pool*. In other words, you may gain money in an absolute sense due to the value of the assets in the pool going up, but because of impermanent loss, you might have gained more money by just holding.

In order for it to be worth it to provide liquidity, the trading fees you earn (plus any additional yield incentives you might be getting) must be enough to counteract the impermanent loss that will happen to you.

First I'll tell you when impermanent loss happens, and then I'll explain what it is.

Impermanent loss happens whenever the price of the two assets in the pool **change relative to each other**. The "relative to each other" part is really important. If the two assets go up in perfect lockstep together, or down together, or stay still together, then there is no impermanent loss. But if one goes up or down while the other doesn't move, or they go up or down together, but by different amounts, or (worst of all) one goes up while the other goes down, then you will experience impermanent loss.

Note that this means **providing liquidity for stable pairs like USDC/DAI** means you are basically **not exposed to impermanent loss** *or* price movements, assuming pegs hold. This is why those pools tend to offer far less reward.

Also note that stable/non-stable pairs are not necessarily more safe from impermanent loss that non-stable/non-stable pairs. With the latter, if the two assets tend to go up together and down together, then that pair will likely experience less impermanent loss than a stable/non-stable pair.

To understand what impermanent loss actually is, we need an example. Let's imagine two scenarios: one in which you just hold 1 ETH and 2000 USDC, and one in which you provide 1 ETH and 2000 USDC to a liquidity pool. Assume that the price of ETH is 2000 USDC at the time you provide to the pool, and that you own 10% of the pool. Thus, the pool must have 10 ETH and 20,000 USDC in it. Assume for simplicity that no other liquidity provider adds or removes liquidity to the pool while you are in it.

Now let's say the price of ETH in the eyes of the world spikes to 3000 USDC. This would cause arbitrage traders to quickly buy up 2 ETH from our pool for 2000 USDC each, because that would mean the pool now contains 8 ETH and, 24,000 USDC, which is a ratio of 3000 : 1. This means that our pool is now in agreement with the rest of the world, so we have found equilibrium, and there are no more arbitrage opportunities.

Now let's say you pull your liquidity. You own 10% of the pool, so you get 10% of the 8 ETH, and 10% of the 24,000 USDC. So, you get 0.8 ETH and 2400 USDC. Since ETH is worth 3000, the total value of your assets is (0.8 * 3000) + 2400 = $4800.

As for our holder: they still have 1 ETH and 2000 USDC, for a total of $5000.

So, **we lost $200** to impermanent loss by providing liquidity. Hopefully the trading fees and yield incentives were enough to offset that so that we are actually rewarded for taking more risk than holding.
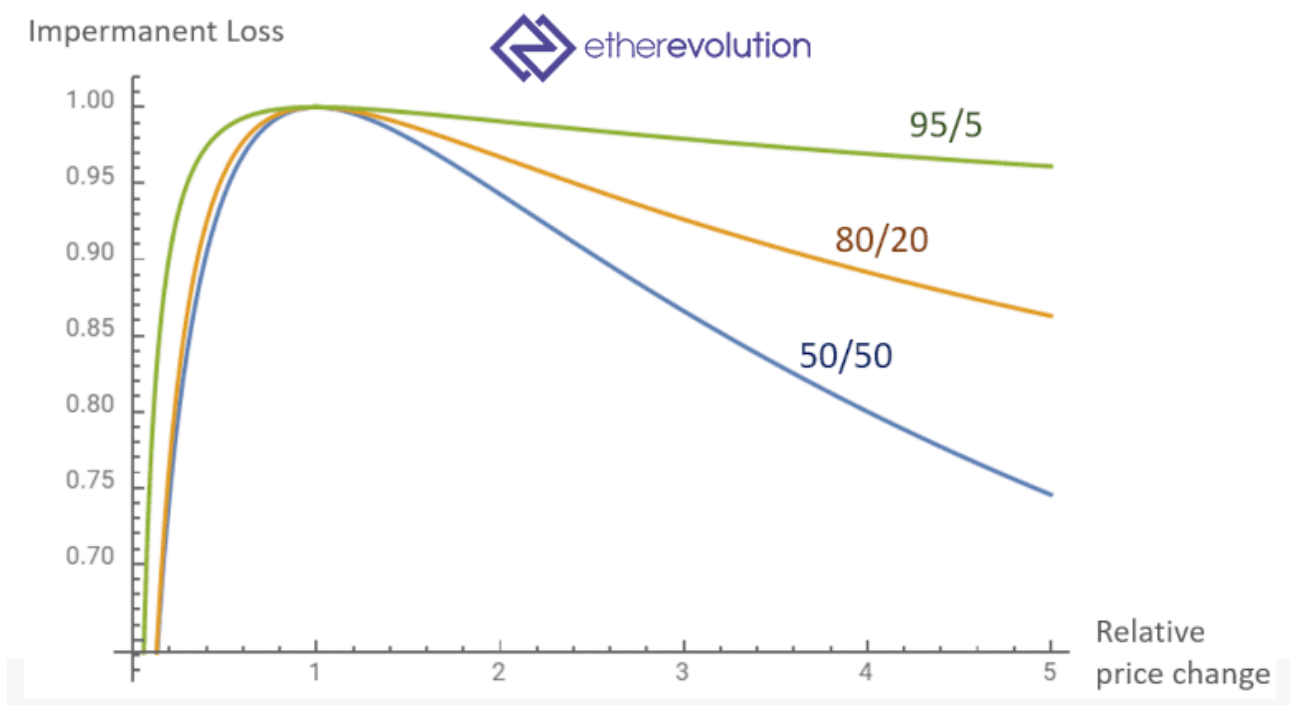
6 Closing thoughts

This should give you a good grounding in the core concepts of DeFi. We covered the **impermanent loss** that happens to **liquidity providers** when they supply to **liquidity pools**, which are the central idea of **AMMs**, which are the smart contracts at the heart of **DEXes**, which are the centerpiece of **DeFi**.

DeFi contains a lot more than just decentralized exchanging. Some of the other things you can do are **borrow and lend, insure your assets, make synthetic assets, trade derivatives, use dynamic yield optimizers, and take out flash loans** (imagine being able to anonymously (and with no collateral) borrow hundreds of millions dollars for about 15 seconds, which you can use to try to generate profits in any way you want, and if you can't pay it back (with interest) by the end of the smart contract, then it reverses everything you did and you never borrowed the fortune in the first place).

The playground that is DeFi is full of many wonders. You could learn about it seemingly forever. Hopefully this post has given you a good launch pad to explore the rest of this world by teaching you the fundamentals of DeFi's most integral idea: **decentralized trading**.