.

A Mathematical Introduction to The Constant Product and Concentrated Liquidity AMMs

Dr.Nick
Oxcacti
August 2023

In this article, we will summarize the main components and mathematics of the *constant product* protocol, as well the *concentrated liquidity* variation. Additionally, for each of these trading protocols, we derive expressions for the *impermanent loss*, which is the primary type of risk to which liquidity providers are exposed. This intended audience of this article is for those who know little to nothing about automated market making, but it can also serve as reference for prerequisite material needed for later articles. It will be divided into 7 sections:

- Section 1: The Constant Product Protocol
- Section 2: How Do We Take Fees?
- Section 3: Impermanent Loss
- Section 4: Concentrated Liquidity
- Section 5: Taking Fees in Concentrated Liquidity
- Section 6: Impermanent Loss In Concentrated Liquidity
- Section 7: The ILF

In a typical exchange, a centralized authority must be present to coordinate trades. In a decentralized exchange, the central authority is replaced by some kind of protocol that determines how trades are to be made. Such protocols are called Automated Market Makers (AMMs). There are numerous kinds of AMMs, but the most common (and elegant) is known as *Constant Product*, originally popularized by Uniswap.

To understand the constant product AMM, we begin by considering a liquidity pool, which is just a contract that contains two (or more) asset types, which we refer to as Token X and Token Y. These assets belong to liquidity providers (LPs) and the pool exists to facilitate trades between the two types of tokens. The amount of each token present in the pool (often called the *reserves*) can be viewed as a coordinate in the x/y plane (Figure 1).

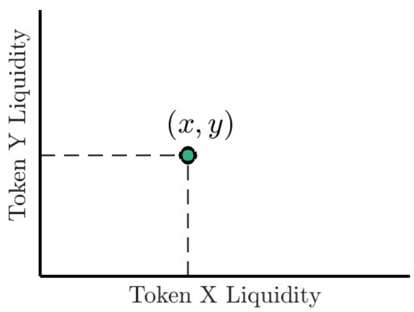

Figure 1. Token reserves represented in x/y plane

When a trade occurs, either some amount Δx enters the pool in exchange for some amount Δy leaving the pool, or vice versa. Thus, in the x/y plane, every trade either moves the state of the pool *down and to the right*, or it moves *up and to the left*, as in Figure 2.

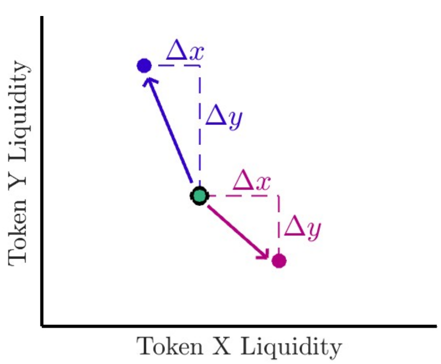

Figure 2. Possible trade directions

Now, suppose for example, that a sequence of trades are made with Δx *entering* the pool. If a constant amount of Δy were to be removed from the pool for each trade, then the pool would eventually run out of Y tokens. Instead, we could imagine that the amount of Δy removed should diminish for each subsequent trade (as illustrated in Figure 3). This makes sense; in such a sequence, the demand for token Y is increasing, and so the exchange rate relative to X should reflect this.

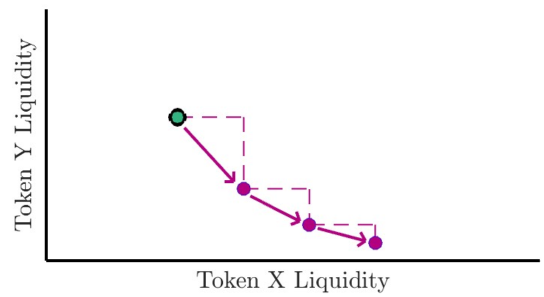

Figure 3. Diminishing returns for repeated trades

Motivated by these observations, the *constant product* protocol is a simple rule that requires the that state of the pool (x, y) must always satisfy the equation $xy = k$ for some constant k. This equation determines a *hyperbola* in the x/y plane, as illustrated in Figure 4.

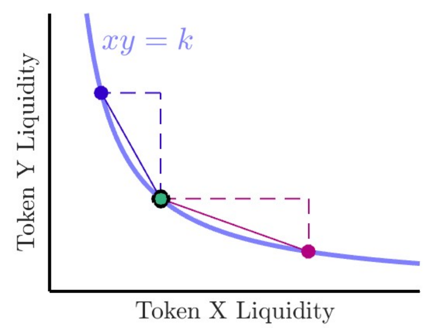

Figure 4. The constant product hyperbola $xy = k$

The state of the pool begins on this hyperbola, and when an amount Δx or Δy is added to the pool, a corresponding amount is removed from the pool that will ensure the state of the pool *remains* on the hyperbola. This simple rule satisfies our previous observations, and also possesses some elegant features.

For each trade, the price of token X relative to Y is, by definition, the *ratio* of Δy and Δx, and this is just the absolute value of the slope of the *secant* line between the points before and after the trade. However, for small trades, this price is approximately given by the slope of the *tangent* line. The curve $xy = k$ has the very special (and elegant) property that the slope of the tangent line (or rather, the absolute value of the slope) at any point is equal to the *ratio* of the coordinates x and y (Figure 5).

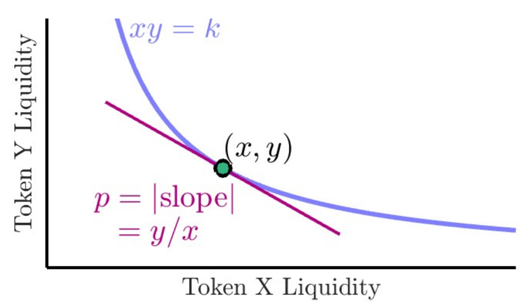

Figure 5. The spot price as the slope of the tangent line

This fact is easily verified; if we write the equation for the hyperbola as $y = k/x$ and take the derivative $dy/dx = -k/x^2$, then after substituting in $k = xy$ and simplifying, this results in $dy/dx = -y/x$. Clearly the slope of the tangent is negative, but the absolute value of this tangential slope can be thought of as the *instantaneous price*, and we will denote it by p:

$$p = y/x. \tag{1}$$

It is often conventional (and convenient) to write our constant k as L^2, and so the equation defining the hyperbola becomes

$$L^2 = xy. \tag{2}$$

We can use equation (1) and equation (2) to write the coordinates (x, y) in terms of p and L. For example, one can divide equation (2) by (1) and take the square root to obtain x. Alternatively, one multiplies (2) by (1) and takes the square root to obtain y:

$$x = L/\sqrt{p} \tag{3}$$

$$y = L\sqrt{p}. \tag{4}$$

Now, in Figure 2, we can see conceptually how trades are determined by the hyperbola, but how does one actually determine the outgoing quantity? Suppose, for example, that an incoming amount Δx is being paid into the pool. There will be a corresponding outgoing quantity Δy, and the state of the pool will be moved from (x, y) to $(x + \Delta x, y + \Delta y)$. By definition, both of these states must be such that their coordinates multiply to give k (being on the curve $xy = k$). Thus,

$$xy = k = (x + \Delta x)(y + \Delta y). \tag{5}$$

This can be expanded, simplified and solved for Δy:

$$\Delta y = -\frac{y\Delta x}{x + \Delta x}. \tag{6}$$

Conversely, if an amount Δy was paid into the pool, then one can solve (5) for Δx to obtain

$$\Delta x = -\frac{x\Delta y}{y + \Delta y}. \tag{7}$$

Thus, equations (6) and (7) tell us how to compute the outgoing quantities, given the current state of the pool, and given some incoming quantity.

Lastly, we note that the value of L is determined by summing over the assets of all LPs. Thus, its value will change as LPs add or remove their assets from the pool. However, this is unrelated to the rules for trading, and so it will not be part of our discussion.

Section 2: How Do We Take Fees?

In order to incentivize LPs to provide their liquidity, a fee is taken from each transaction. For the constant product protocol, the standard way to do this is to take a fee from each incoming quantity. However, rather than immediately deliver this fee to the LPs, the fee is *left in the pool*, as extra liquidity.

To accomplish this mathematically, we introduce a parameter γ which measures the fee size. Specifically, $1 - \gamma$ is the fraction of the incoming quantity that is to be taken as a fee. For example, if $\gamma = 0.997$, then the fee size is 0.3% (a typical value). Now suppose again that an incoming amount Δx is to be paid into the pool. This incoming quantity can be split into the amount $\gamma\Delta x$ and $(1 - \gamma)\Delta x$, the latter being a fee (Figure 6).

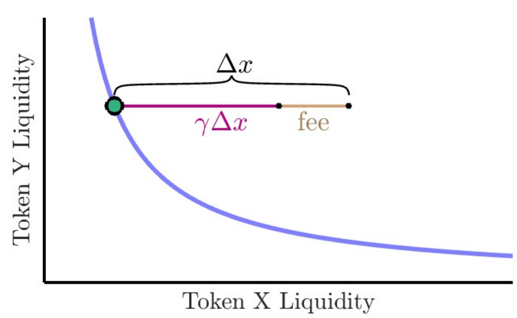

Figure 6. Separating a fee from the incoming quantity

We use this *reduced* input $\gamma\Delta x$ to compute the corresponding output quantity Δy to be received from the trade. Thus, for an incoming quantity Δx, we will modify our formula for the output (6), by choosing to use $\gamma\Delta x$ as our input amount instead:

$$\Delta y = -\frac{y\,(\gamma\Delta x)}{x + (\gamma\Delta x)}. \tag{8}$$

Similarly, for an input Δy, formula (7) becomes

$$\Delta x = -\frac{x\,(\gamma\Delta y)}{y + (\gamma\Delta y)}. \tag{9}$$

We use (8) or (9) to compute the output, but rather than deliver the fee to the LPs, we deposit the *entire* input amount into the pool, ending up at a state that is *above* the constant product curve (see Figure 7). Since each LP is entitled to a certain consistent fraction of the pool, then their assets grow with each transaction.

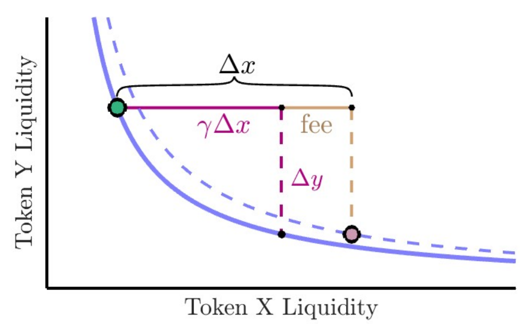

Figure 7. Storing the fee internally

One notes that because of this slow and steady accumulation of extra liquidity, the name *constant product* is technically a misnomer, as the 'constant' k (and therefore L) slightly increases with each trade.

When LPs provide liquidity to a constant product trading pool, they expose themselves to a type of risk we call *impermanent loss* (IL). This is the opportunity cost of providing liquidity, defined as the fractional difference between the value that the LP can currently claim minus the value they would otherwise be able to claim if they had not entered the pool:

$$\text{IL} := \frac{\begin{pmatrix}\text{current value}\\ \text{held in pool}\end{pmatrix} - \begin{pmatrix}\text{current value had}\\ \text{they stayed out}\end{pmatrix}}{\begin{pmatrix}\text{current value had}\\ \text{they stayed out}\end{pmatrix}} \tag{10}$$

To better articulate this quantity, we will need to express value in terms of one (arbitrary) denomination, which will generally be chosen to be the Y token denomination. Specifically, because the price p converts an amount of X tokens into a corresponding amount of Y tokens, then the total value of the pair (x, y), expressed in the Y token denomination, is given by $px + y$.

Now, let us suppose that the LP initially provides an amount of liquidity (x_0, y_0). At some later time, the liquidity in the pool has changed to (x, y), with a current price of $p = y/x$, and the total value held by the LP is therefore $px + y$. However, had the LP *stayed out* of the pool, they would still hold the amount (x_0, y_0), which would now constitute a total value of $px_0 + y_0$. Thus, we can re-express (10) as

$$\text{IL} = \frac{(px + y) - (px_0 + y_0)}{(px_0 + y_0)} \tag{11}$$

$$= \frac{(px + y)}{(px_0 + y_0)} - 1. \tag{12}$$

Next, we rewrite every x and y using (3) and (4):

$$\text{IL} = \frac{\left(p(L/\sqrt{p}) + L\sqrt{p}\right)}{\left(p(L/\sqrt{p_0}) + L\sqrt{p_0}\right)} - 1. \tag{13}$$

We divide top and bottom by $L\sqrt{p_0}$, giving

$$\text{IL} = \frac{2\sqrt{p/p_0}}{(p/p_0 + 1)} - 1. \tag{14}$$

If we define the price ratio r by

$$r := p/p_0, \tag{15}$$

then we finally arrive at

$$\text{IL} = \frac{2\sqrt{r}}{r+1} - 1. \tag{16}$$

We note that canceling L between (13) to (14) is not quite correct. As we've seen, the value of L will slowly grow over time via fees, and so the value of L in the numerator and denominator are not actually the same (this will be addressed in Section 7). But we proceed here as though there were *no fees*.

By finding a common denominator and simplifying, expression (16) can be written in the alternative form

$$IL = -\frac{(1-\sqrt{r})^2}{1+r}. \tag{17}$$

From this expression, we can make two very important observations:

- The IL is **always negative** (or zero). This is manifestly obvious from expression (17).
- The IL is **invariant** under a change from r to $1/r$. This can be seen by direct calculation - we substitute $1/r$ into the expression for IL, and retrieve (17):

$$\frac{(1-\sqrt{1/r})^2}{1+1/r} = \frac{(1-\sqrt{1/r})^2}{1+1/r}\left(\frac{r}{r}\right) = \frac{(\sqrt{r}-1)^2}{r+1}$$

We plot expression (17) for IL v.s. r below (Figure 8):

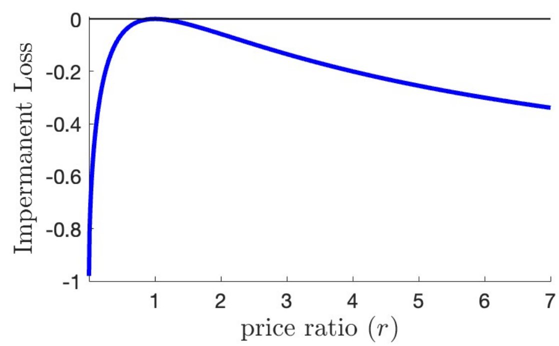

Figure 8. Impermanent loss v.s. price movement

The non-positive nature of IL is clear from Figure 8, but the symmetry that we noted in our second observation is less obvious. The invariance between r and $1/r$ means that, for example, the IL is the same whether the price *doubles* or *halves*. This can be more readily seen if the IL is plotted on a logarithmic axis (see Figure 9 below). We see, for example, that whether the price grows by a factor of 2 or 1/2, the resulting IL is -0.057 (or -5.7%).

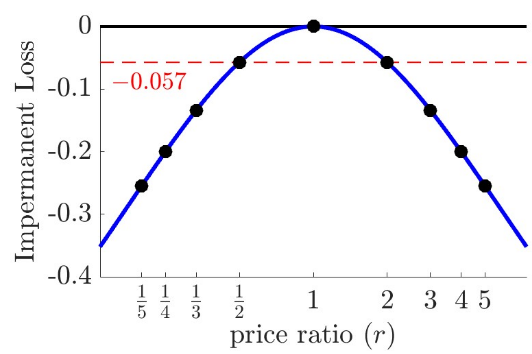

Figure 9. Impermanent loss on a logarithmic axis

Thus, the severity of the IL is irrespective of the *direction* of the price movement, and it is only non-negative if the price returns to the initial value (where IL=0).

Section 4: Concentrated Liquidity

Next we discuss a variation of the constant product protocol known as *concentrated liquidity* (introduced in Uniswap V3). We begin by noting that each point on the constant product hyperbola corresponds to a price p, and thus a range of prices $[p_a, p_b]$ corresponds to a unique curve *segment*. The essential idea of concentrated liquidity is that we may apply liquidity differently in different price ranges. Towards that end, we imagine dividing up the constant product hyperbola into numerous curve segments, and then shifting each segment (down and to the left) so that it fits snugly against the coordinate axes (illustrated in Figure 10).

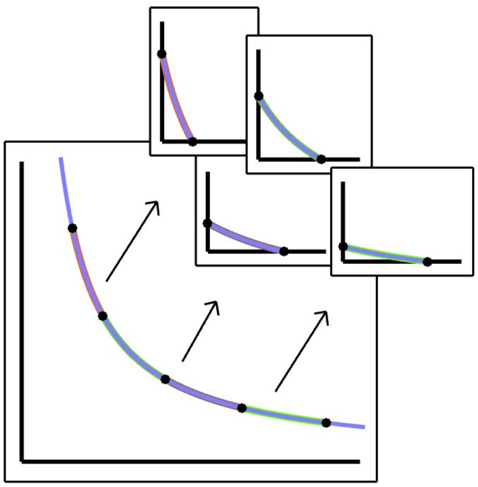

Figure 10. Hyperbola split into (shifted) curve segments

Consider a particular price range $[p_a, p_b]$ and a liquidity factor L corresponding to a *virtual* curve $xy = L^2$. Using (3)-(4), we find that to shift the curve segment down to the coordinate axes, the appropriate shift amounts are $L\sqrt{p_a}$ and $L/\sqrt{p_b}$ (down and to the left).

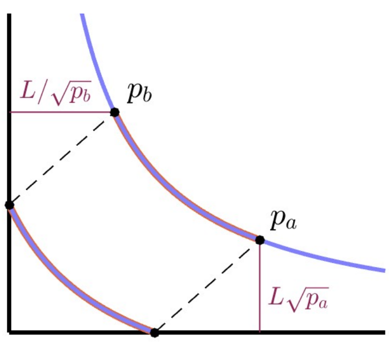

Figure 11. Shifting a segment to the coordinate axes

We may now think of this shifted curve segment as establishing a trading protocol for trades that occur *within* the range $[p_a, p_b]$. The *virtual* token reserves in (3)-(4) are shifted by the amounts shown in Figure 11, giving us expressions for the *actual* token reserves:

$$x = L/\sqrt{p} - L/\sqrt{p_b} \qquad (18)$$

$$y = L\sqrt{p} - L\sqrt{p_a}. \qquad (19)$$

We notice that, along a given curve segment, the underlying geometry that determines the rules for trading is unchanged whether the segment is shifted or not (Figure 12):

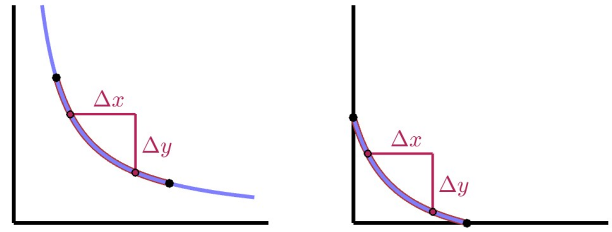

Figure 12. Same trade, shifted or not

Thus, by shifting the curve segment, we are able to conduct the same trades, but with much *less* liquidity (at least, constrained to this particular price range).

Next we imagine a collection of LPs, each one of them making an individual choice of a price range $[p_a, p_b]$ in which to provide liquidity, along with a chosen liquidity factor L. The complete collection of these prices $\{(p_a, p_b)\}$ (coming from *all* LPs) can be listed in ascending order and given an index (say, i), thus partitioning our price axis along a set of prices $\{p_i\}$, called *ticks*. Between each pair of consecutive ticks, we have a range $[p_i, p_{i+1}]$. On this range, the total liquidity that is available to facilitate trades is that which is provided by any LPs whose chosen price range $[p_a, p_b]$ overlaps with $[p_i, p_{i+1}]$. Thus the liquidity factor L for the range $[p_i, p_{i+1}]$ will just be the *sum* of the L's from each relevant LP. This is illustrated in Figure 13:

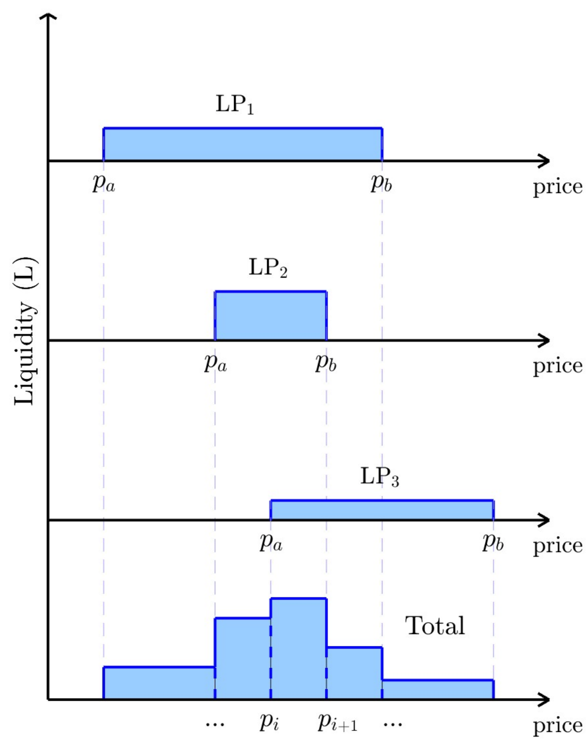

Figure 13. Superposition of LP positions

More explicitly, the value of L for the range $[p_i, p_{i+1}]$ is

$$L = \sum \begin{pmatrix} \text{Any } L \text{ from an LP such} \\ \text{that } p_a \le p_i \text{ and } p_{i+1} \le p_b \end{pmatrix} \tag{20}$$

Now, to conduct a trade within the range $[p_i, p_{i+1}]$, rather than specifying a desired input amount, as in formulas (6)-(7), for concentrated liquidity it is cleaner to specify the *price movement* (i.e. give a target price). Explicitly, if the current price p and the target price p' are both in the same current interval $[p_i, p_{i+1}]$, then formulas (18)-(19) tell us what the change in x and y will be as we move from p to p':

$$\Delta x = L\Big(1/\sqrt{p'} - 1/\sqrt{p}\Big) \tag{21}$$

$$\Delta y = L\Big(\sqrt{p'} - \sqrt{p}\Big), \tag{22}$$

where L is the quantity described in (20).

However, if the price movement for a trade spans multiple ranges, then in general we do the following:

- Execute the trade *as far as we can* within the current price range $[p_i, p_{i+1}]$, using the current L
- Carry on into the next range (updating the value of L), and execute the trade as far as we can, iterating over as many price ranges as needed to execute the entire trade

One can see that the capital efficiency previously noted comes at a computational cost.

Lastly, we note that in practice, Uniswap V3 uses the index i slightly differently; it actually ranges over a discrete set of *allowable* ticks, while the specific ticks chosen by LP positions are distinguished as *active* ticks. For the sake of the exposition here, this difference is immaterial.

Section 5: Taking Fees In Concentrated Liquidity?

To take fees in a concentrated liquidity pool, the situation is somewhat different. If we attempt to store the fees internally, as we did for the constant product protocol, we will run into issues of proper fee allocation and expensive book keeping. Instead, the current standard practice is to store the fees *externally* from the pool. In other words, a small fee is taken from the incoming quantity and then stored separately, to be delivered to the LPs directly. Nevertheless, some care must be taken in order to ensure that the fees will be given only to the *relevant* LPs, and done so *pro rata*.

As before, we begin with a fee parameter γ, and we split the input quantity, say Δx, into two pieces; $\gamma\Delta x$ and $(1-\gamma)\Delta x$. The former quantity is fed into the protocol outlined previously, while the latter quantity is stored as an external fee. However, it is natural that an LP ought to only receive fees for transactions that occur over the range in which they provide liquidity. To accomplish this, there is a clever (if somewhat opaque) solution.

First, a global fee quantity f_g is defined which keeps a running total of the fees earned over all transactions. Next, for each tick p_i, a quantity $f_o(i)$ is defined, which is often misleadingly referred to as the 'fees earned *outside* of the i^{th} tick' (of course, it is meaningless to speak of a region *outside* of a single price p_i). The true meaning of $f_o(i)$, which depends on the current price p, can be stated as follows:

$$f_o(i) = \left\{ \begin{matrix} \begin{pmatrix} \text{fees from trades} \\ \text{starting below } p_i \end{pmatrix} & \text{if } p > p_i \\ \begin{pmatrix} \text{fees from trades} \\ \text{starting above } p_i \end{pmatrix} & \text{if } p < p_i \end{matrix} \right\} \tag{23}$$

In other words, one can describe $f_o(i)$ as the 'fees earned on the *other* side of the i^{th} tick' (where the notion of the *other* side is relative to the *current* price p).

With a little thought, one confirms that each time the state of the current price p transitions *across* the tick p_i, we must update $f_o(i)$ according to

$$f_o(i) = f_g - f_o(i), \tag{24}$$

For example, supposing we currently have $p < p_i$, then (23) tells us that $f_o(i)$ measures the cumulative fees taken from *above* p_i. If we next move across this tick so that we now have $p > p_i$, then (24) tells us to subtract $f_o(i)$ from the *total* cumulative fees f_g, giving us the cumulative fees taken from *below* p_i, in agreement with the definition in (23). How we initialize $f_o(i)$ is actually immaterial, because the only thing relevant to each LP is the *change* in $f_o(i)$ from one time to another.

Now, using $f_o(i)$, one can compute the cumulative fees earned *above* and *below* the tick p_i by the following:

$$f_{below}(i) = \begin{cases} f_o(i) & \text{if } p > p_i \\ f_g - f_o(i) & \text{if } p < p_i \end{cases} \tag{25}$$

$$f_{above}(i) = \begin{cases} f_g - f_o(i), & \text{if } p > p_i \\ f_o(i) & \text{if } p < p_i \end{cases} \tag{26}$$

Thus, for an LP providing liquidity in the price range $[p_a, p_b]$, we can compute the amount of earned fees corresponding to this range by subtracting the fees *outside* of the range from the global fee tally:

$$f_{[p_a,p_b]} = f_g - \big[f_{below}(i_a) + f_{above}(i_b)\big]. \tag{27}$$

In this way, we can ensure that for each fee taken, only the *relevant* LPs receive them.

To ensure that the fees are delivered *pro rata*, the fee variables are actually stored in units of 'fees per *one unit* of liquidity'. Thus, to compute the appropriate reward for a given LP, we simply multiply this quantity by the LP's liquidity factor L.

Working out an expression for the impermanent loss in a concentrated liquidity pool is more complicated than it was for the constant product pool. However, we first note that for an LP providing liquidity in the range $[p_a, p_b]$, with a chosen liquidity factor L, we can treat this range as a stand alone process. Because the liquidity factors add linearly on each price range, the presence of other LPs is immaterial. In other words, if the price moves from p_0 to p, the assets of any particular LP transform in a uniquely specified way that does not depend on any other LP positions.

Thus, we begin by fixing a price range $[p_a, p_b]$, and a liquidity constant L. If the current price p is still within this interval ($p_a \le p \le p_b$), then we once again invoke expression (12), but now we substitute (18) and (19) in for x and y:

$$\begin{aligned} IL &= \frac{(px+y)}{(px_0+y_0)} - 1 \\ &= \frac{p\big(L/\sqrt{p} - L/\sqrt{p_b}\big) + \big(L\sqrt{p} - L\sqrt{p_a}\big)}{p\big(L\sqrt{p_0} - L/\sqrt{p_b}\big) + \big(L\sqrt{p_0} - L\sqrt{p_a}\big)} - 1. \end{aligned} \tag{28}$$

We will divide top and bottom by a factor of $L\sqrt{p_0}$ and simplify to find

$$IL = \frac{2\sqrt{p/p_0} - (p/p_0)\sqrt{p_0/p_b}\big) - \sqrt{p_a/p_0}}{(p/p_0)(1-\sqrt{p_0/p_b}) + (1-\sqrt{p_a/p_0})} - 1. \tag{29}$$

Analogous to our definition (15) of $r = p/p_0$, we define

$$r_a := p_a/p_0, \tag{30}$$
$$r_a := p_b/p_0. \tag{31}$$

Then (29) can be written as

$$IL = \frac{2\sqrt{r} - r/\sqrt{r_b} - \sqrt{r_a}}{r(1-1/\sqrt{r_b}) + (1-\sqrt{r_a})} - 1. \tag{32}$$

Finally, if we simplify (32) with a common denominator, then (after simplifying and canceling terms) we find

$$IL = \frac{-\big(1-\sqrt{r}\big)^2}{(1-\sqrt{r_a}) + (1-1/\sqrt{r_b})r}, \quad (p \in [p_a, p_b]) \tag{33}$$

One can compare expression (33) with expression (17). In particular, we note that in the limit where $r_a \to 0$ and $r_b \to \infty$, we ought to retrieve the constant product formula, and indeed one can check that (33) becomes (17) in this limit.

However, we are not done. We must still consider the case where p moves *outside* of the range $[p_a, p_b]$. We begin with $p < p_a$. As Figure 11 suggests, when the price moves below p_a, all of the assets in our price range will have been converted to the X token only, where the amount of which is given by $L/\sqrt{p_a} - L/\sqrt{p_b}$. Thus, we may repeat our previous calculation, but now we simply have

$$\begin{aligned} IL &= \frac{(px)}{(px_0+y_0)} - 1 \\ &= \frac{p\big(L/\sqrt{p_a} - L/\sqrt{p_b}\big)}{p\big(L/\sqrt{p_0} - L/\sqrt{p_b}\big) + \big(L\sqrt{p_0} - L\sqrt{p_a}\big)} - 1. \end{aligned} \tag{34}$$

Once more, after dividing through by $L\sqrt{p_0}$ and writing our expression in terms of the r variable, we have (after some simplification)

$$IL = -\frac{\big(1-1/\sqrt{r_a}\big)r + \big(1-\sqrt{r_a}\big)}{(1-\sqrt{r_a}) + (1-1/\sqrt{r_b})r}, \quad (p < p_a). \tag{35}$$

Similarly, if $p > p_b$, then only the Y token will be left in the subpool, and thus we can write

$$\begin{aligned} IL &= \frac{(y)}{(px_0+y_0)} - 1 \\ &= \frac{L\sqrt{p_b} - L\sqrt{p_a}}{p\big(L\sqrt{p_0} - L/\sqrt{p_b}\big) + \big(L/\sqrt{p_0} - L/\sqrt{p_a}\big)} - 1. \end{aligned} \tag{36}$$

Again, after dividing through by $L\sqrt{p_0}$, writing the result in terms of the r variable, and simplifying, we arrive at

$$IL = -\frac{\big(1-1/\sqrt{r_b}\big)r + \big(1-\sqrt{r_b}\big)}{(1-\sqrt{r_a}) + (1-1/\sqrt{r_b})r}, \quad (p_b < p). \tag{37}$$

If we stitch together our expressions (33), (35) and (37), we can obtain a complete picture of IL over the whole range of possible price movements. We plot this below in Figure 14 as a function of r, with $r_a = 0.5$ and $r_b = 2$, superimposed over our original expression for IL in the constant product pool, from expression (17).

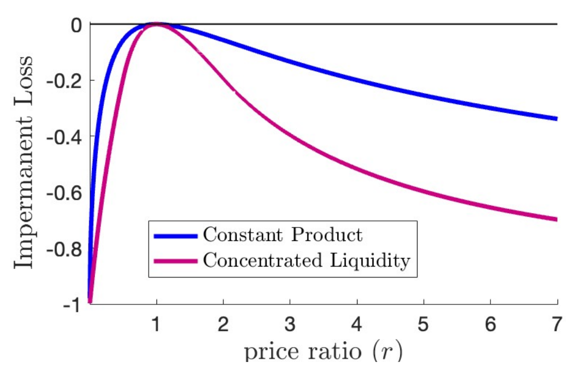

Figure 14. Impermanent loss in concentrated liquidity

The immediate conclusion is that impermanent loss is *more severe* in the concentrated liquidity pool. Any deviation from $p = p_0$ (i.e. $r = 1$) results in a more dramatic IL value in the concentrated liquidity pool than it otherwise would in a constant product pool.

As was previously mentioned, the calculation for impermanent loss in (17) is not quite accurate in practice, due to the non-constant (increasing) nature of the liquidity constant k as a consequence of taking fees. This means that the impermanent loss is not actually strictly negative. Now, this makes sense of course, for *if* the probability of incurring loss was guaranteed, then there would be no incentive to LP. However, rather than modifying our calculation for IL, we will define a *new* quantity that includes the effect of fees (and we will leave the definition of IL as it is). This new quantity, which will be fundamental for any analysis to follow, will be called the ILF, i.e. the *impermanent loss with fees*. Returning to our definition of the IL,

$$\mathrm{IL} := \frac{\begin{pmatrix}\text{current value}\\ \text{held in pool}\end{pmatrix} - \begin{pmatrix}\text{current value had}\\ \text{they stayed out}\end{pmatrix}}{\begin{pmatrix}\text{current value had}\\ \text{they stayed out}\end{pmatrix}} \tag{38}$$

we augment it slightly to include the fees:

$$\mathrm{ILF} := \frac{\begin{pmatrix}\text{current value}\\ \text{held in pool}\end{pmatrix} + \text{Fees}\Big) - \begin{pmatrix}\text{current value had}\\ \text{they stayed out}\end{pmatrix}}{\begin{pmatrix}\text{current value had}\\ \text{they stayed out}\end{pmatrix}} \tag{39}$$

After all, the fees must be included to truly asses the current success of an LP. We must be careful, however, to precisely articulate what is meant with the '+Fees' term. As we have seen, in the constant product protocol, the fees are stored internally in the pool in the form of extra pool liquidity, and so these fees are counted in (39) implicitly via the growth of L. However, as we have seen, in the concentrated liquidity protocol the fees are stored externally outside of the pool. In this case, we count these external fees by simply adding them to the existing pool value. To summarize, then,

$$\left(\begin{matrix}\text{current value}\\ \text{held in pool}\end{matrix} + \text{Fees}\right) = \left\{\begin{matrix} \begin{pmatrix}\text{value in pool,}\\ \text{given rise in } L\end{pmatrix} & \begin{matrix}\text{if fees stored}\\ \text{internally}\end{matrix} \\ \\ \begin{pmatrix}\text{value in pool}\\ + \text{ external fees}\end{pmatrix} & \begin{matrix}\text{if fees stored}\\ \text{externally}\end{matrix}\end{matrix}\right\} \tag{40}$$

We will consider the ILF in more detail in later articles, both mathematically, and for numerical simulation. For the moment, let us run a brief simulation to illustrate some important features of the IL and ILF.

In Figure 15 below, we have the results of a simulation of 10,000 trades, with a fee size of 0.3%. We plot the IL and the ILF, along with the series of underlying market prices that are responsible for generating these IL/ILF values (initial price is 1.0).

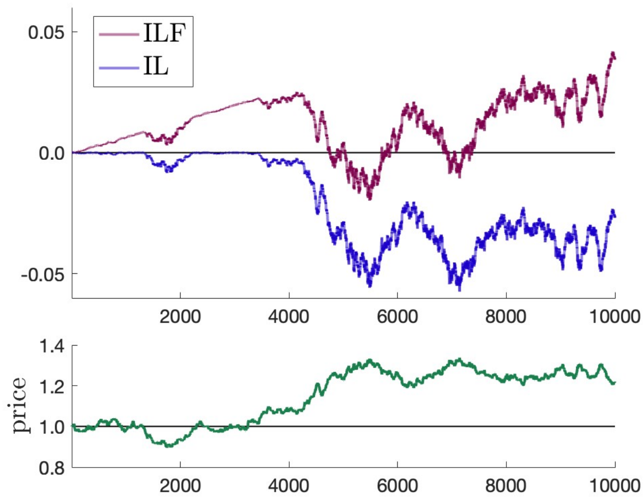

Figure 15. The IL and ILF for a given simulation

Let us make some observations:

- The IL is never positive, as expected.
- The IL returns to near zero whenever the price approaches the initial price, and it falls off from zero whenever the price moves far above *or* below the initial price.
- The ILF can be positive. Having definitional overlap with the IL, we see that the ILF has a similar trajectory as the IL, but it is buoyed into the positive by the accumulation of fees.
- Of course, the ILF can be negative too (the fees do not always make up for the IL). In fact, with more volatile price movements, the ILF can be entirely negative.

In future articles, we will run many more simulations like this one.