What is the color of Uniswap bleed?

> Version v0.1.1, 21 August 2020
>
> Stefan Loesch (skloesch at gmail dot com)

Intro

TODO

Invariants and trading prices

The key Uniswap formula is the invariant relationship $k = x * y$. What this means that agents *trading with the pool* (as opposed to agents *providing liquidity to the pool*) can freely choose their location on this line when trading with the pool. To give an example: if the current holdings of the pool are x_0 and y_0 (in their own arbitrary *number of tokens* units) and someone chooses to provide dx tokens of type X to the pool then in return they'll get dy tokens of type Y where $x_1 * y_1 = k$, $x_1 = x_0 + dx$ and $y_1 = y_0 + dy$.

What we are really interested is the *marginal pool trading price* P which we consider as function of dx, and which is defined as

$$P(dx) = -\frac{dy}{dx}$$

The minus sign appears because whenever tokens of type X are contributed to the pool then tokens of type Y are withdrawn and vice versa. The signs of dx and dy are from the perspective of the pool, ie a positive number means the token is contributed, and a negative number means it is withdrawn.

First a number of easy to prove identities that can come helpful along the way. First the *percentage change relationship*

$$\frac{dy}{y_1} = -\frac{dx}{x_0}$$

and

$$\frac{dy}{y_0} = -\frac{dx}{x_1}$$

Note the respective indices — the dy term is *backwards*. Resolving x_1 in the equation above yields

$$\frac{dy}{dx} = -\frac{y_0}{x_0 + dx}$$

which we can rewrite as

$$\frac{dy}{dx} = -\frac{y_0}{x_0} \cdot \frac{1}{1 + \frac{dx}{x_0}}$$

If we define the *pool price* $P_t = y_t / x_t$ for $t \in \{0, 1\}$ then we have

$$P = -\frac{dy}{dx} = P_0 \cdot \frac{1}{1 + \frac{dx}{x_0}} = P_0 \cdot (1 - \frac{dx}{x_0} + \cdots)$$

where we used $1/(1+x) = 1 - x + \cdots$. The second term is the *slippage term* and it essentially says that the marginal trading price is the current pool price adjusted for the relative size of the trade: if you want to trade *1%* of the pool's assets then the price gets *1%* worse.

Choice of invariants

Hypothesis. In this example the invariant function is $f(x, y) = x * y$. The hypothesis is that the exact form of the invariant function does not matter, as long as it is strictly decreasing and it asymptotically meets the two axes.

Justification. The asymptotics are important as the pool can never be allowed to run out of tokens. This means that even for very large dx the dy must be less than the amount of tokens available y_0. Similarly, one can not buy more tokens dx than are available in the pool x_0. The function must be decreasing because otherwise at least at some points it would be possible with withdraw tokens of both types without breaking the invariant.

Note that *for small trades* the slippage term goes to unity, so for small trades we find

$$P = -\frac{dy}{dx} = P_0 = \frac{y_0}{x_0}$$

ie for small trades the marginal price equals the pool price.

Within an active market

Arbitrage condition

So what happens if we connect this pool to a wider market? Let's assume that at time 0 there is (at least) one market participant who has access to a firm price p meaning that he can exchange X vs Y in any quantity and direction at a a price ratio of p. This trader will trade with the pool whenever she can make a profit. The only way there is no profit to the made is if the marginal price for small trades (ie ignoring slippage) P_0 equals the market price p, ie we have

$$p = P_0 = \frac{y_0}{x_0}$$

In other words, the arbitrage trader will trade with the pool until the above condition holds.

As a reminder, x, y is the number of tokens of type X, Y in the pool. For more clarity of the exposure let's assume that v_x is the value of X in some numeraire, for example USD. So xv_x is the dollar value of the x tokens in the pool. We know that the relative price $v_x/v_y = p$ by our definition of p (the unit of p is "Y per X"), therefore $yv_y = yv_x/p$ is the value of tokens of type Y in the pool. From the arbitrage condition above we know that $p = y/x$, and therefore we have

$$yv_y = xv_x$$

In other words, **when the pool is fully arbitraged against a deep external market then the monetary value of both pool constituents X and Y is the same**.

Market moves

Now we are looking what happens if markets move. Let's assume we start at time 0 with a market price $p_0 = v_x/v_y$ and an arbitraged pool, ie

$$p_0 = P_0 = \frac{y_0}{x_0}$$

The first thing to note is that whoever provides liquidity to the pool is exposed to the combined market risk of X and Y, ie if that liquidity provider accounts in another numeraire (and the notation v_x, v_y implies this) then if both X and Y increase / decrease by 10% then the pool value increases / decreases by the same value.

The more interesting part however is when we have relative price movements between X and Y. To study this we are placing ourselves in the numeraire X, ie we will have $v_x = 1$ throughout and all volatility will be caused by the price changes of Y, ie v_y.

So let's assume that the market jumps instantaneously from p_0 to p_1 and that the pool is still at its p_0 arbitraged state. As a reminder, total pool value before the jump was $2x_0$ (because both parts of the pool have equal value, and the value of 1 X is 1 by choice of numeraire) and after the jump it is

$$x_0\left(1 + \frac{p_0}{p}\right)$$

This intuitively makes sense: p is measured in "Y per X" so if p goes up this means the value of Y goes down, and as the pool is long Y the value goes down. The opposite holds as well - if p decreases the value of Y and therefore of the pool goes up. This is however not the end of it: remember the pool is always willing to trade on its invariant line $x * y = k$ and as the pool is currently out of kilter it is profitable doing so.

Calculations

The calculations are not hard but a bit confusing, so we'll go through step by step. As a reminder we have (1) $x_0y_0 = x_1y_1$ from the *invariant*, and (2a) $y_0/x_0 = p_0$ and (2b)

$y_1/x_1 = p1$ from the *arbitrage conditions*. Multiplying (1) with (2b) yields

$$y_1^2 = p_1 x_0 y_0$$

We now write $p_1 = p_0 + dp$ and we can transform the above equation to

$$y_1^2 = \left(1 + \frac{dp}{p_0}\right) y_0^2$$

and, if we take the square root

$$y_1 = \sqrt{1 + \frac{dp}{p_0}} y_0$$

Using (1) we get the equivalent equation for x

$$x_1 = \frac{x_0}{\sqrt{1 + \frac{dp}{p_0}}}$$

To simplify the formulas we temporarily use $c = \sqrt{\cdots}$ as the square root term and we compute

$$\frac{dy}{dx} = \frac{y_1 - y_0}{x_1 - x_0} = \frac{(c-1)y_0}{(\frac{1}{c} - 1)x_0}$$

Using the identity $\frac{c-1}{1/c-1} = -c$ and resubstituting c we find

$$\frac{dy}{dx} = -\sqrt{1 + \frac{dp}{p_0}} p_0 = -\sqrt{p_0 p_1}$$

where for the last identity we simply use $p_1 = p_0 + dp$ and move the p_0 into the square root.

Conclusion. What we have seen here is that when the pool has been arbitraged at the price point p_0 and prices move to p_1 then the pool allows the arbitrageur to trade at a price $\sqrt{p_0 p_1}$, ie the (geometric) average of p_0 and p_1, whilst she is at the same time able to trade in the market at a price of p_1. The arbitrage ratio is therefore

$$\frac{\sqrt{p_0 p_1}}{p_1} = \sqrt{1 - \frac{dp}{p_1}}$$

To give a numerical example, if dp/p_1 is 10% then that ratio is 94.8 meaning that she can make a risk free profit of 5.2% on the amount she can trade with the pool.

Pool bleed and negative Gamma

We have seen above that the pool executes its trades suboptimally: when coming from price p_0 and being a price p_1 then the pool allows execution at some average of p_0 and p_1 (in this case the geometric average; for other invariant functions this will differ, but it always will be an average).

One might think that this is just a suboptimal but still winning strategy rather than an outright loss, but this is not quite right as we are concerned with the market making profits of the pool here. Pool investors have a long exposure to both of the assets in the pool, and in order to cancel this they will have to go short outside of the pool. However, after the above market move the execution outside the pool they will be at p_1, and if inside the pool they execute at the average $avg(p_0, p_1)$ they'll *loose* money after every adjustment.

As we calculated above, the relative price loss is

$$\sqrt{1 - \frac{dp}{p_1}} - 1 = 2\frac{dp}{p_1} + \cdots$$

where we used $\sqrt{1+x} = 1 + 2x + \cdots$.

That relative price loss needs to be applied to a scale to give the value loss, and the scale in this case is the traded value dy which is

$$dy = \sqrt{1 + \frac{dp}{p_0}} - 1 = 2\frac{dp}{p_0} + \cdots$$

We multiply those two terms together and, observing that p_0 and p_1 only differ by a term dp which can be ignored in our Taylor expansion, we find that the bleed $b(dp)$ after a price move dp is

$$b(dp) = \frac{4}{p_0^2} dp^2$$

Smooth random process

We now want to examine what happens under frequent rebalancing, ie if the arbitrageurs jump in whenever there is a chance to do so. This is a realistic assumption in the presence of multiple, non-colluding arbitrageurs. This is essentially prisoners' dilemma: whilst everyone would gain by waiting until dp is large (remember the gain is quadratic in dp) any individual will be better off by just taking profits if and when they arise.

We first assume that our market process is random but smooth, meaning that $dp \propto dt$ when we make our time steps dt smaller and smaller. As to the latter, we consider a macroscopic time interval T and we divide it into N equal pieces so that $dt = T/N$. Frequent rebalancing is in this case expressed by the limit $N \to \infty$.

In this limit our dp gets smaller. More precisely it scales with $1/N$. The value change is quadratic in dp and therefore scales with $1/N^2$. The number of pieces increases with N . So overvall in the smooth case our aggregate bleed is proportional to

$$N \times \frac{1}{N^2} = \frac{1}{N} \to 0$$

Conclusion. In case of a smooth price process the arbitrage "bleed" (ie the money lost to arbitrageurs) goes to zero with instantaneous rebalancing.

Brownian motion process

Now we assume that our price process is a Brownian motion. As a reminder, a Brownian motion is defined as

$$dW = \mu dt + \sigma dZ$$

where dZ is a standard Brownian motion with $E[dZ] = 0$ and $E[dZ^2] = dt$. This latter relationship is often slightly sloppily denoted as $dZ^2 = dt$ and we find that for dW the equivalent

$$dW^2 = \sigma^2 dt$$

which can easily be "proven" by using the definition above and ignoring all terms that are higher order than dt (ie only the term dZ^2 survives).

This is really the defining property of a Brownian motion, and it is what distinguishes it from a smooth random process: in the former, quadratic changes are in dt^2 and therefore disappear when going granular (ie $N \to \infty$; see above). A Brownian motion is much more jerky and rugged, to the extent that dW^2 is of order dt (and deterministic) which means it has macroscopic effects even in the granularity limit $N \to \infty$.

To repeat the calculation above, let's assume that dp is a logarithmic Brownian motion with volatility parameter σ like in the Black Scholes model, ie $dp^2 = p^2 \sigma^2 dt$. The aggregate bleed to arbitrageurs in this case is

$$\sum^N \frac{4}{p^2} dp^2 = \sum^N \frac{4}{p^2} p^2 \sigma^2 dt \to 4\sigma^2 T$$

where as above $dt = T/N$ and the last relation follows in the limit $N \to \infty$

Jump process

A jump process is a process where most of the time nothing happens, but from time to time there are violent (technically: discontinuous) motions, aka "jumps". For the most common jump process, in every small time interval dt the probability of a jump occurring is dt, and for non-overlapping time periods the jumps are statistically independent. In discrete time those can be modelled with a Bernoulli distribution (at every time interval dt there is a typically very small probability p of a jump, and the jumps are all independent) and in continous time this converges against the Poisson distribution and process. The mathematics of this is well known, but outside of the scope of this paper as it is not very interesting here.

It is worth however at this stage to compare Brownian motion and jump processes as those are, in some respect, diametrically opposite. The Brownian motion is very jittery and violent at a small scale. Regardless of how closely one looks, the direction always abruptly changes and the trajectory never has a tangent. However, the trajectory is continuous (no jumps) and most of the movements are very very small. The larger the scale the less important the random effects. A Brownian motion is, on most trajectories, dominated by the drift term. Also a squared Brownian motion is deterministic ($dZ^2 = dt$) because most of the small distance jittery moves are "absorbed" by the flat region around the origin of the square function.

Jump process on the other hand are nice, smooth and calm most of the time — in fact in the most simple cases simply nothing happens at all. However, when something happens, oh boy! Whilst a Brownian motion is continuous everywhere a jump process is discontinuous at the jump points. Details now depend the exact form of the jump parameters. In the easiest case a jump is just a multiplicative or additive constant. However, it can be any distribution one wants as it is somewhat mix-and-match: there is one dynamics for *when* a jump happens, and there is another one for what happens *in case of a jump*.

Whilst we do not want to look at the time dynamics here it is interesting to analyse what happens in a jump. We remember from above that the bleed associated with a jump dp is

$$b(dp) = \frac{4}{p_0^2} dp^2 \propto \left(\frac{dp}{p_0}\right)^2$$

ie the bleed is proportional to the square of the percentage move. Note a key difference to Brownian motion is that in the latter case the bleed was deterministic and continuous whilst in the case of a jump process it is stochastic and occasional.

In any case, what we do here is do look at the expected bleed

$$E[b(dp)] = E\left[\left(\frac{dp}{p}\right)^2\right]$$

We find that the expected bleed is closely related to the variance of the returns and equal to it if dp/p is a Martingale, ie has zero drift.

For reasonably nice jump behaviour this variance exists, in which case we are by an large back to our Brownian motion case provided we are looking at it from a long enough timescale. In the long run, central limit theorem will ensure that the combined jumps converge to something like a Gaussian distribution.

However, there are many fat-tailed distributions out there whose variance is infinity. What this means is that, in the long run, there will be a jump that will wipe out the pool.

Non-technical summary and conclusion

In this paper we have looked at the Uniswap protocol which is an automated market making protocol currently implemented on the Ethereum blockchain. This protocol is based on so-called *"liquidity pools"* which are smart contracts containing two different tokens, and who are willing to trade with everyone who comes along according to a pre-determined formula.

This formula is based on the invariant $x * y = k$ where x, y are the respective token amounts (in their own native units) of the tokens X, Y held in the contract, and k is a constant. The key property of the Uniswap contract is that the invariant function is its *indifference function* as well - the contract will engage into any trades X vs Y that respect the invariant (for the avoidance of doubt this ignores fees the contract might charge).

The first result we found that in presence of a deep and liquid market that is exchanging X vs Y at a price p arbitrageurs will make sure that the monetary value of the X tokens in the contract equals the monetary value of the Y tokens as for any other pool composition the price the pool offers allows an arbitrage opportunity. We did not prove this, but the arbitrage result is a generic result that holds for many different invariant functions (not all lead to equal monetary value howoever. In fact, our hypothesis is that every function that is (a) strictly decreasing, and (b) has the x and y axis' as asymptotics is valid as invariant.

The second result we found was that a pool that is currently arbitraged at a price p_0 will offer to trade at a price $\sqrt{p_0 p_1}$ if the market price moves to p_1. Note that this is an arbitrage opportunity for others as they can trade in the market at the price p_1. This arbitrage opportunity is by design as it ensures that the pool will be in its correct state after each move. Coming back to our hypothesis, the result with the *geometric average* is not generic. However, in the generic case the price offered by the pool will be an average of p_0 and p_1 and the arbitrage persists.

We then made a key observation that is generally overlooked in the discussion about the benefits of contributing ones assets to a Uniswap pool. It is important to separate the gains and losses from the underlying position from the benefits of the arbitrage strategy. To take a step back: if someone is providing liquidity to a Uniswap pool they are forcibly long the two pool assets and therefore are exposed to the volatility in the value of those assets. In order to understand the benefits of contributing to a pool we need to strip out the effect of that long position and need to look at the pool effect only.

In order to do this adjustment we assume that the liquidity provider run an equivalent short strategy outside the pool. More precisely, they start out with a short position of equal monetary size at the current price p_0 and after every market movement they adjust their position to be again at equal monetary size *at the then prevailing price* p_1. This last sentence in italics is key: to maintain their market neutral position they execute the trades on the short side at the then prevailing market price $p1$; however, the pool trades with the market at a price that is an average of p_0, p_1 which is by construction and design more favourable than p_1.

The above means that at every finite adjustment that fully hedged strategy suffers a value bleed. This is a well known phenomenon in finance, and specifically option pricing theory, which is known as *"negative Gamma"* where the position bleeds value at every adjustment of the hedge.

We found that the value bleed is quadratic in dp, meaning that a twice bigger move leads to four times the bleed. This is important as it exaggerates the impact of big moves, and it diminishes the impact of small moves. On the face of it this suggest that whilst it exists for finite moves we can make the bleed arbitrarily small by changing the adjustment

frequency (and we argued that in a competitive markets arbitrageurs would not be able to collude and wait until the moves are bigger to reap more value).

We looked at this proposition that the bleed disappears as long as the position is adjusted frequently enough for three distinct cases: a smooth random process, a Brownian motion, and a jump process. We found that our intuition held only in the case of the smooth random process, which unfortunately is not a realistic description of market dynamics.

In case of a **Brownian motion** market dynamics (the same that is being used in the Black Scholes option pricing universe) we found that the bleed of the Uniswap pool is proportional to $\sigma^2 T$ where σ is the volatility of returns, and T is the accrual period. We also found that in case of the Brownian motion that bleed is deterministic, just like the bleed on a constant-Gamma position in a Black Scholes world is deterministic.

We also looked at the **jump process** market dynamics, which in some respect is the opposite of a Brownian dynamics: for the Brownian motion the main uncertainty is very small movements over a very short timescale, but over a longer scale the uncertainty is contained. For the jump process on the other hand the uncertainty is restricted to specific moments in time — most of the time the process is smooth and nothing happens. However, when things are happening they are violent. At best there is a finite discontinuity in the process, but nothing prevents the jumps from being extremely violent and *fat tailed*. We found that if the jump distribution has infinite variance (as is usually the case for fat tailed distributions) then the expected bleed is infinite.

As we said above, the difference between a jump distribution and a Brownian motion is that in the former the bleed is stochastic. This means that for a certain period of time we can expect to see non-catastrophic jumps that, within measurement boundaries, can be interpreted of being as the benign *finite-variance* type. However, at one point we can expect a massive jump (and/or a series thereof) that completely wipes out the value to the liquidity providers.

Conclusion

We have shown that by design Uniswap liquidity providers suffer a bleed whenever the pool is adjusted after market movements. This bleed is a necessary component for Uniswap to function as it attracts the arbitrage traders who ultimately keep the pool up to date. For most liquidity providers this bleed is hidden within the volatility of their systemic long position. However, when looking at it on a market neutral basis with all the token risk hedged out the bleed is a net cost to the liquidity providers.

A priori this does not have to be a problem: this bleed is very similar to that suffered by option traders, and for which the option premium is the compensation. Just like in the option trading case it is important that the bleed is covered by the market making fees.

This however leads to a number of important conclusions.

- Firstly, market making fees are not all *profit* as they need to be adjusted for bleed.
- Secondly, as the bleed is proportional to the square of the volatility of the price process σ^2, a change in that volatility will lead to a change in profitability for the market makers unless the fees are adjusted dynamically
- Finally, in the presence of fat-tailed jumps, notably jumps with infinite variance, there is no level of market making fee that can compensate, and liquidity providers will ultimately face ruin.

References

1. https://ethresear.ch/t/improving-front-running-resistance-of-x-y-k-market-makers/1281
2. https://hackmd.io/@HaydenAdams/HJ9jLsfTz#🦄-Uniswap-Whitepaper