On the mathematics of Uniswap prices

Kairon Labs Autumn 2020

1 Uniswap v2 :

Uniswap is a protocol on Ethereum for swapping ERC20 tokens without the need for buyers and sellers to create demand. It does this via an equation that automatically sets and balances the value depending on how much demand there is. Uniswap's main distinction from other decentralized exchanges is the use of a pricing mechanism called the "Constant Product Market Maker Model". In short, this means that in a pool with two tokens, the product of the amount of tokens and the token price for each token should always be equal such that the total value of both assets in the pair is equal at every point in time i.e. 50/50.

2 Problem

It is easily seen that buying x tokens for a given amount of ETH will influence the price of the token, so that the next buyer will pay more for the same amount of tokens. Looking at it from a different perspective, buying tokens with ETH will reduce the amount of tokens and increase the amount of ETH. As a result, less tokens are equal in value to more ETH, meaning the price of the token will increase.

2.1 Programming solution

The above problem is easily solved by a program that buys a set amount of tokens step by step, updating the prices and amounts and re-uses the output as input for the next step. Although creating a while loop yields a correct solution to the problem, an analytical solution will be given below.

2.2 Analytical solution

Let's call t_e, p_e, t_t, p_t respectively the amount of ETH[1], the price of ETH, the amount of tokens and the price of the token. Let us evaluate a random token and ETH pair. We will denote an amount of tokens bought by x, so that $t_e(x)$ is the amount of ETH after buying x tokens (or alternatively swapping ETH for the token). The *step*, in analogy to the programming solution, will be denoted by ϵ. We assume ϵ to be an infinitesimal change in the amount of tokens bought. The amount of ETH in the pool after buying $x + \epsilon$ tokens is equal to the amount of tokens after buying x tokens, plus ϵ times the ratio of the price of the token after buying x tokens and the price of ETH, or in symbols :

$$t_e(x + \epsilon) = t_e(x) + \frac{\epsilon \cdot p_t(x)}{p_e}$$

The second term on the right hand side can be understood by the following reasoning. $\epsilon \cdot p_t(x)$ denotes the price for an infinitesimal amount of tokens bought at price $p_t(x)$, with $p_t(x)$the price of the token when x tokens have been bought from the initial situation, i. e. $p_t(0) = p_{t,0}$. Dividing this by the price of ETH gives an amount of ETH, corresponding to the extra ϵ tokens bought.
From the above equation we get

$$\frac{t_e(x + \epsilon) - t_e(x)}{\epsilon} = \frac{p_t(x)}{p_e} \Rightarrow t'_e = \frac{p_t(x)}{p_e}$$

Where the prime denotes a derivative with respect to x. Following the same analogy

$$\frac{t_t(x + \epsilon) - t_t(x)}{\epsilon} = -1 \Rightarrow t'_t = -1 \tag{1}$$

Which is obvious, since the amount of tokens after buying ϵ tokens minus the amount of tokens before buying ϵ tokens, divided by ϵ, gives $-\epsilon/\epsilon = -1$.

[1]Note that ETH can be easily replaced by a stablecoin such as USDC, USDT, DAI etc.

We are interested in the evolution of the token price as a function of x, i.e. how many tokens are bought. The goal of this derivation is to obtain an analytical function which will give us the evolution of the price of the token as a function of tokens bought.
The essence of Uniswap's automated market maker is, as stated in the introduction $p_t(x) \cdot t_t(x) = p_e \cdot t_e(x)$. To calculate the evolution of the price, mathematical steps ϵ have to be taken. A variation ϵ in the amount of tokens bought gives

$$\begin{aligned} p_t(x+\epsilon) = \frac{t_e(x+\epsilon)\cdot p_e}{t_t(x+\epsilon)} &= \frac{t_e(x+\epsilon)\cdot p_e}{t_t(x)+\epsilon} \\ &= \frac{(t_e(x)+\epsilon\cdot\frac{p_t(x)}{p_e})\cdot p_e}{t_t(x)+\epsilon} \\ &= \frac{t_e(x)\cdot p_e+\epsilon\cdot p_t(x)}{t_t(x)+\epsilon} \\ &= \frac{t_t(x)\cdot p_t(x)+\epsilon\cdot p_t(x)}{t_t(x)+\epsilon} \end{aligned}$$

From which follows (remember : $log\left(\frac{a}{b}\right) = log(a) - log(b)$)

$$\begin{aligned} p_t(x+\epsilon) &= p_t(x)\cdot\frac{t_t(x)+\epsilon}{t_t(x)-\epsilon} \\ \Rightarrow \frac{p_t(x+\epsilon)}{p_t(x)} &= \frac{t_t(x)+\epsilon}{t_t(x)-\epsilon} \\ \Rightarrow log\left(\frac{p_t(x+\epsilon)}{p_t(x)}\right) &= log\left(\frac{t_t(x)+\epsilon}{t_t(x)-\epsilon}\right) \\ \Rightarrow log(p_t(x+\epsilon)) - log(p_t(x)) &= log(t_t(x)+\epsilon) - log(t_t(x)-\epsilon) \\ &= log\left(t_t(x)(1+\frac{\epsilon}{t_t(x)})\right) - log\left(t_t(x)(1-\frac{\epsilon}{t_t(x)})\right) \\ &= \cancel{log(t_t(x))} + log\left(1+\frac{\epsilon}{t_t(x)}\right) \cancel{-log(t_t(x))} - log\left(1-\frac{\epsilon}{t_t(x)}\right) \end{aligned}$$

If we let $log(p_t(x)) = y(x)$ we can write, taking into account $a\cdot log(x) = log(x^a)$:

$$\frac{y(x+\epsilon)-y(x)}{\epsilon} = log\left(\left(1+\frac{\epsilon}{t_t(x)}\right)^{1/\epsilon}\right) - log\left(\left(1+\frac{\epsilon}{-t_t(x)}\right)^{1/\epsilon}\right)$$

In the limit of $\epsilon \to 0$, we get $\lim_{\epsilon\to 0}\left(\frac{y(x+\epsilon)-y(x)}{\epsilon}\right) = y'(x)$. The limit calculation of the right hand side is a little more complicated. Firstly, we recognize that the logarithmic function is a continuous function, meaning we can bring the limit inside. Then, we introduce $n = \frac{1}{\epsilon}$ and calculate

$$\lim_{n\to\infty}\left(log\left(\left(1+\frac{1}{n\cdot t_t(x)}\right)^n\right)\right) = log\left(\lim_{n\to\infty}\left(\left(1+\frac{1}{n\cdot t_t(x)}\right)^n\right)\right)$$

where we recognize that $\lim_{n\to\infty}\left(1+\frac{1}{n\cdot t_t(x)}\right)^n = e^{1/t_t(x)}$. This means

$$y'(x) = log\left(e^{1/t_t(x)}\right) - log\left(e^{1/t_t(x)}\right) = \frac{1}{t_t(x)} - \frac{1}{-t_t()}$$

This finally gives us

$$y'(x) = \frac{2}{t_t(x)} \tag{2}$$

Equation (1) and (2) now give us the following set of equations

$$\begin{cases} t'(x) = -1 & \Rightarrow t_t(x) = -x + t_{t,0} \\ y'(x) = \frac{2}{t_t(x)} & \Rightarrow y(x) = \int_0^x \frac{2dx}{t_{t,0}-x} \end{cases}$$

The latter equation can be solved, yielding

$$\begin{aligned} y(x) &= \int_0^x \frac{2dx}{t_{t,0} - x} \\ &= -2\left(log(t_{t,0} - x) - log(t_{t,0})\right) \\ &= 2log(t_{t,0}) - 2log(t_{t,0} - x) \end{aligned}$$

remember, we put $y(x) = log(p_t(x))$, resulting in

$$log(p_t(x)) = log\left(\frac{t_{t,0}^2}{(t_{t,0} - x)^2}\right)$$

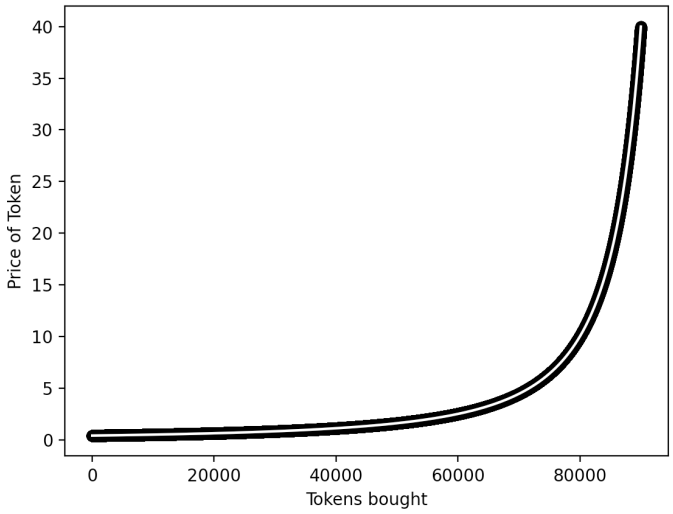
This then finally results in an analytical equation, providing us with the price of the token $p_t(x)$ as a function of tokens bought/sold x.

$$\boxed{p_t(x) = \left(\frac{t_{t,0}}{t_{t,0} - x}\right)^2 \cdot p_{t,0}}$$

where $t_{t,0}$ and $p_{t,0}$ represent the initial amount of tokens and the initial price of the token, respectively. This formula allows us to calculate the price impact of selling/buying x tokens, with a given amount of initial tokens in the pool $t_{t,0}$ and the initial price of the token $p_{t,0}$. Note that the total amount of ETH and the price of ETH in the pool is not in the equation, but will be needed to calculate the price of the token, of course.

3 Case study

Let's apply the above formula to a simple example. Assume a pool on Uniswap, containing 100 ETH and 100.000 tokens. For the sake of simplicity, the price of 1 ETH = 400 USD.

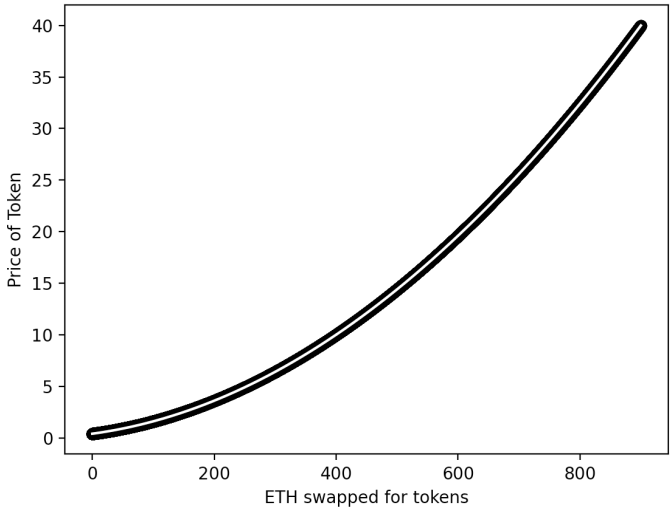


The black data points are generated by a Python script as described before. The white line is the boxed formula above. We can see that the formula exactly matches the computational data. Given the price of ETH, an initial amount of tokens : init_token, an initial amount of ETH : init_eth and the initial price of the token, calculated as $\frac{\text{init\_eth} \cdot \text{eth\_price}}{\text{init\_token}}$, a simple Python script to plot the above graph is given below.

This three rule program replaces the while loops as used before.

```
plot_token = [i for i in range(int(init_token*.9))]
plot_y = [init_token**2/(init_token-e)**2 * init_price for e in plot_token]
plot_eth = [(init_token-plot_token[i])*plot_y[i]/eth_price -init_eth for i in range(len(plot_token))]
```

Firstly, the x-axis is defined as ranging from 0 to 0.9 times the initial amount of tokens. If the total amount of tokens would be used, the price would go to infinity. The list plot_token represents the amount of tokens bought from the pool. Next, a list is generated where the boxed formula on the previous page is applied to the values in the list of x values, giving us a list with the price of the token at a given point on the graph. As an extra option, the list plot_eth can be generated with the purpose of generating the amount of ETH to be put in the pool, instead of the amount of tokens to be bought, which is most likely of more interest to the user. To do so, the amount of tokens that are bought from the pool is subtracted from the initial amount of tokens in the pool. This is then multiplied by the price of the token at that point, divided by the price of ethereum, yielding an amount of ETH that has to be put in the pool to achieve a certain token price as shown below



It can be seen that the plot where the x axis in ETH value behaves more linearly as opposed to the graph where the amount of tokens is used. This can be readily explained by the fact that the value of the token rises as the amount of tokens bought rises, meaning more ETH has to be put in the pool to buy the same amount of tokens, giving the x axis a more 'stretched' character in the second plot.

Furthermore, this equation can be extended to x values below 0, meaning ETH is withdrawn from the pool and tokens are added to the pool.