Element.fi Ecosystem Simulation

Alex

September 2021

1 Constant Power Sum Invariant AMM

As Niemerg, Robinson, and Livnev (2020) indicates, most popular protocols for swapping tokens on Ethereum use pooled liquidity provision and set their prices automatically based on a formula that specifies some fixed relation between the reserves of the two assets, e.g $xy = k$, where x and y are the pool's reserves of the two assets. However, the constant product AMM used by Uniswap, Balancer etc. does not conform to the inherent nature of fixed yield tokens. One of the examples is that the principal token should be priced at its face value when it comes to its maturity. Using a constant product AMM might cause capital inefficiency and LPs would suffer from arbitrage opportunities inside the liquidity pool. So Element.fi decides to incorporate a time parameter and adopt the *constant power sum invariant* AMM as their market making strategies:

$$x^{1-t} + y^{1-t} = k \tag{1}$$

where y represents the reserves of the PT tokens, x represents the reserves of the base asset (such as Dai), and t represents the time to maturity, where t is normalized so that $0 \leq t \leq 1$.

As Equation (1) obviously shows, when it approaches the maturity, the AMM acts more like a stable pair. The *constant power sum invariant* AMM could also allow for more price discovery at the beginning period of the term.

2 Price Discovery & Slippage

Before diving deeper in to the Element.fi ecosystem, we'd better figure out how price discovery works in the *constant power sum invariant* AMM and how to parametrize the AMM so that we could have a dynamic strategy to optimize the slippage and price discovery.

2.1 Pricing of PT

$$
\begin{aligned}
dx^{1-t} + dy^{1-t} &= 0 \\
(1-t)x^{-t}dx + (1-t)y^{-t}dy &= 0 \\
\frac{dx}{dy} &= (\frac{y}{x})^{-t}
\end{aligned} \tag{2}
$$

Thus, we can get $Price_{PT} = (\frac{y}{x})^{-t}$ and $Price_x = (\frac{x}{y})^{-t}$.

2.2 Buy & Sell PTs

Recalling Equation (1), we can use it to demonstrate how the liquidity pool would change in terms of a given input / output amount.

$$(out_r - out_q)^{1-t} + (in_r + in_q)^{1-t} = k$$

where out_r is the reserves of the outgoing asset, out_q is the quantity of the outgoing asset, in_r is the reserves of the incoming asset, in_q is the quantity of the incoming asset, t is the time to maturity and k is a constant.

Calculate Out Given In:

$$CalOutGivenIn(out_r, out_q, in_r, in_q, k, t, k) = out_r - \left(k - (in_r + in_q)^{1-t}\right)^{\frac{1}{1-t}} \tag{3}$$

Calculate In Given Out:

$$CalInGivenOut(out_r, out_q, in_r, in_q, k, t, k) = \left(k - (out_r - out_q)^{1-t}\right)^{\frac{1}{1-t}} - in_r \tag{4}$$

Based on the two scenarios, the slippage would be:

$$Slippage = abs\left(Price_{out} out_q - CalOutGivenIn(out_r, out_q, in_r, in_q, k, t, k)\right)$$
$$Slippage = abs\left(Price_{in} in_q - CalInGivenOut(out_r, out_q, in_r, in_q, k, t, k)\right)$$

2.3 Liquidity Pool Risk Control

2.3.1 Reserve Ratio

For convenience, t is defined as follows:

$$t = \frac{termlength}{365}$$

Considering PT as a fixed income token, the unit price of a PT in terms of APY is:

$$UnitPrice_{PT} = 1 - \frac{APY}{100}t \tag{5}$$

Combining Equation (2) and Equation (5), we can get:

$$\left(\frac{y_r + l_{shares}}{x_r}\right)^{-t} = 1 - \frac{APY}{100}t$$

By assuming $l_{shares} = x_r + y_r$, we can solve the reserve ratio $\frac{x_r}{y_r}$:

$$\frac{x_r}{y_r} = -2\left(\frac{1}{\left(1 - \frac{APY}{100}t\right)^{\frac{1}{t}} - 1} + 1\right) \tag{6}$$

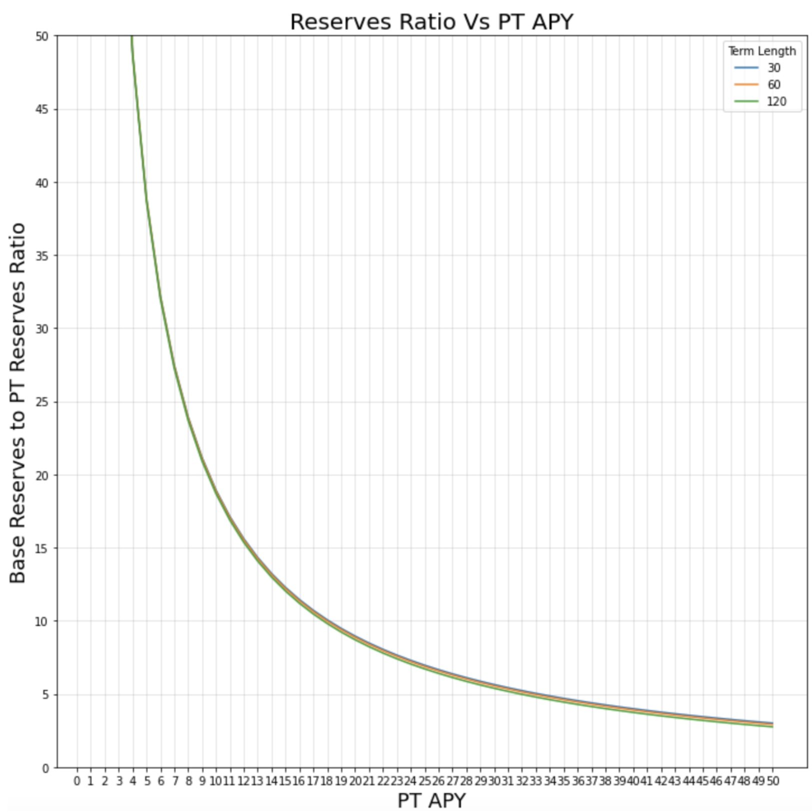

Figure 1: Reserves Ratio Vs PT APY (%)

Based on Figure (3), we can conclude:

- Term (or tranche) length does not have a significant affect on the reserve ratios.
- At a 20% APY, for every 1 PT staked, 9 base asset tokens will be required by LPs. A new parameter (Time Stretch) need to be introduced to the curve to stretch time and reduce the burden on LPs.

2.3.2 Time Stretch

The time stretch is formulated as:

$$t = \frac{termlength}{365 \times t_{stretch}}$$

The reserve ratio would become:

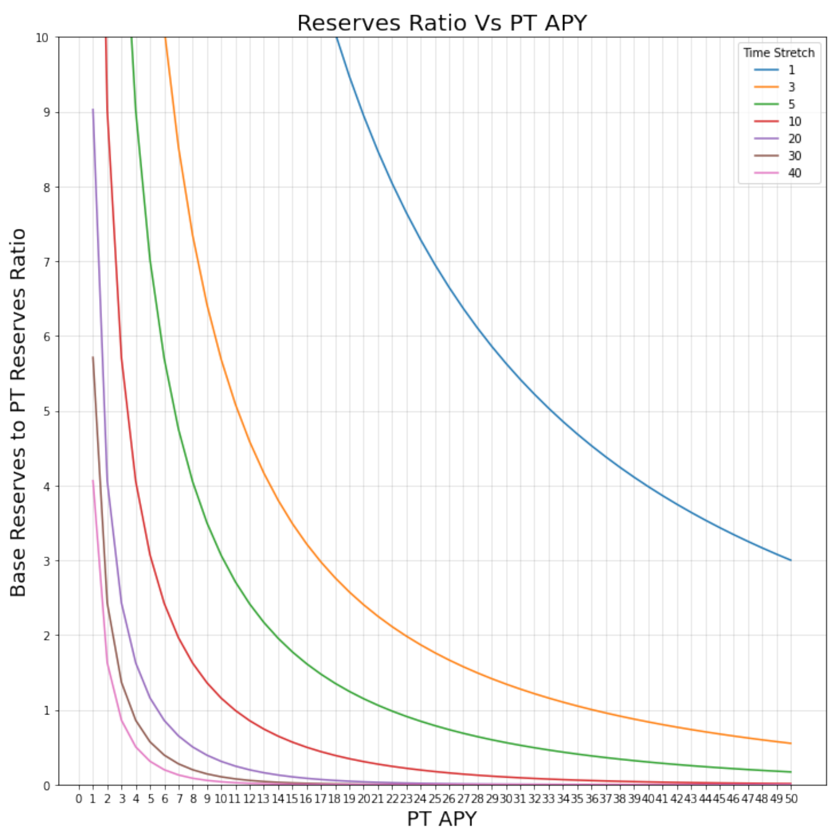

Figure 2: Reserves Ratio Vs PT APY (%)

On an intuitive level, stretching the time parameter forces the invariant to behave more like it would when near maturity.

- The more t is stretched, the more it behaves like it is near maturity and therefore requires more of an imbalance in reserves to reach a higher APY. Thus, large time stretches will see less slippage and the burden of LPs can be reduced.
- The less t is stretched, the more it can enhance price discovery.

2.3.3 Maximum Liquidity

Recapping on equation (3), we should note:

$$k - (in_r + in_q)^{1-\frac{t}{t_{stretch}}} \geq 0$$

The boundary can be determined by defining the largest feasible trade by solving for in_q:

$$in_q \leq k^{\frac{1}{1-\frac{t}{t_{stretch}}}} - in_r$$

If the assumption that $l_{shares} = x_r + y_r$ still holds, the maximum liquidity can be given by:

$$Max_{inputtrade} = k^{\frac{1}{1-\frac{t}{t_{stretch}}}} - (2y_r + x_r) \qquad (7)$$

2.3.4 Optimization for Price Discovery

As mentioned above, $t_stretch$ would have an impact on slippage and price discovery. In order to establish a safe and capital-efficient liquidity pool, Element.fi use the following algorithm to calibrate a dynamic $t_{stretch}$:

$$t_{stretch} = \frac{0.309396}{0.02789 APY_{PT}}$$

Algorithm 1 Find best $t_{stretch}$

for $\{y_r, termlength, APY, t_{stretch}\} \in \Theta$ **do**
    Init x_r and total supply (E.q: 1)
    for $DaysToMaturity = 1, 2, \ldots,$ **do**
        Compute maximum liquidity (E.q: 7)
        Compute fees, average price, slippage, daily APY change, and reserve ratio when the maximum liquidity of PT is sold out (E.q: 3)
    end for
end for
$\{t_{stretch,i}^{APY_j}\} \leftarrow$ For a fixed APY, find every combination that satisfies $ReserveRatio \geq 0.1$ and $MaxAPYChange \geq 50$

Output: $\{\frac{1}{n}\sum_{i=1}^{i=n} t_{stretch,i}^{APY_j}\}$

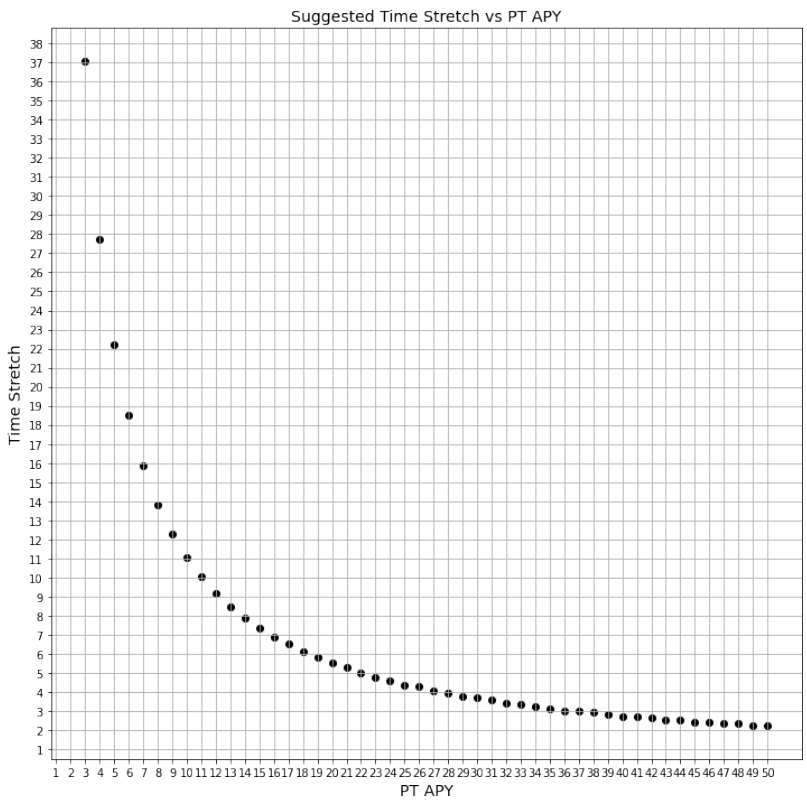

Figure 3: Time Stretch Vs PT APY (%)

3 Simulation

Algorithm 2 Element.fi Ecosystem Simulation

for $\{DailyVolume, Liquidity, Fee\} \in \Theta$ **do**
  Init x_r and y_r through a starting APY and liquidity (E.q: 1)
  Init a random seed
  for $Day = 30, 29, \ldots, 1$ **do**
    $ub \leftarrow DailyVolume \times \log(\frac{DaysToMaturity}{Day})$
    $TodayVolume \leftarrow U(\frac{1}{2}ub, ub)$ ▷ Simulate the waning demand over the lifetime of PT
    while $TodayVolumeDailyVolume$ **do**
      $amount \leftarrow N(\mu, \sigma)$
      Trade $amount$ on a random direction (E.q: 3, 4)
      Accumulate volume, fees and slippage
    end while
  end for
end for

Output: Print out result

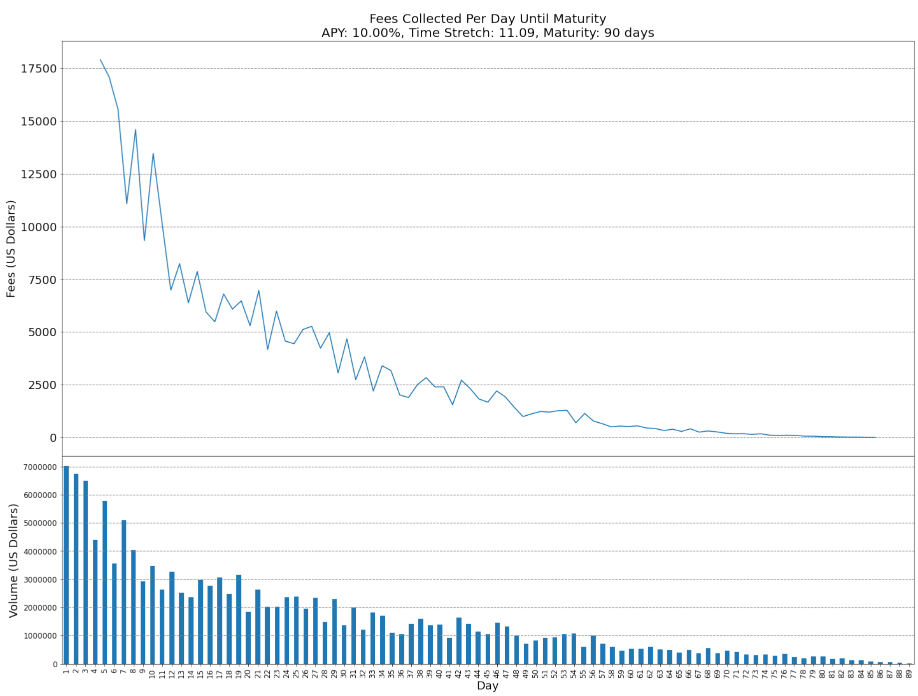

Figure 4: Fees Simulation

References

Niemerg, Allan, Dan Robinson, and Lev Livnev (2020). "YieldSpace: An Automated Liquidity Provider for Fixed Yield Tokens". In: *Retrieved Feb* 24, p. 2021.