Updatable Rate-Scaling for the 3CLP

Steffen Schuldenzucker, Ariah Klages-Mundt*

August 1, 2025

We discuss how to make the 3CLP updatable by adjusting token rates using a special updatable rate provider. This allows the pool to adjust its price range when one or several of its prices are out of range and enables it to be used for volatile pairs like WETH/WBTC/USDC.

1. Setting

Assume that there are three assets with balances $t = (x, y, z)$ with rates $\delta = (\delta_x, \delta_y, \delta_z)$ attached. Fix some parameter $\alpha \in (0, 1)$. The rate-scaled 3CLP constructs a 3CLP curve with respect to the rate-scaled balances $t^\delta := (\delta_x x, \delta_y y, \delta_z z)$ and allows trading along this curve with rate-scaled balances. For example, when swapping x to y, a swap amount Δx would be scaled up to $\Delta x^\delta := \delta_x \cdot \Delta x$, then a corresponding rate-scaled swap amount Δy^δ is computed along the rate-scaled 3CLP curve, and this amount is scaled back to $\Delta y := \Delta y^\delta / \delta_y$.

In the updatable version of this setup, one or more of the rates δ are controlled by a manager contract (the *orchestrator*) and can be updated if the pool is out of range of the current market prices. Assume that we are given oracle prices $p = (p_x, p_y, p_z)$. The quote asset of these prices does not matter since we will be using relative prices exclusively. We will assume in the following WLOG that (1) the numeraire is asset z and the price vector is

$$p = \left(p_{x/z} = \frac{p_x}{p_z}, p_{y/z} = \frac{p_y}{p_z}\right)$$

and we assume (2) that the updatable rate provider controls the rates δ_x and δ_y. If this is not the case, we simply need to rotate our asset names.[1]

We can transform prices between rate-scaled and actual space. Specifically, let

$$p^\delta = \left(\frac{\delta_z}{\delta_x} p_{x/z}, \frac{\delta_z}{\delta_y} p_{y/z}\right)$$

be the rate-scaled vector of oracle prices.

2. Feasible pool prices and equilibrium

We now need to consider what it means for the 3CLP to be “out of range”. Label the rate-scaled (“inner”) pool spot prices offered by the pool $P^\delta = \left(P^\delta_{x/z}, P^\delta_{y/z}\right)$. Note that the third pair price, $P^\delta_{x/y}$, can be left out because the pool’s internal prices are always arbitrage-

*Researchers at Superluminal Labs, a software development company working in the Gyroscope ecosystem. steffen@gyro.finance, ariah@gyro.finance

[1]Note here that the 3CLP is symmetric as well, so we do not need to transform any pool parameters when we rename assets. We could also assume for this exposition WLOG that and $\delta_z = 1$, i.e. asset z does not have a rate provider, but we do not here. Our implementation *does* establish this, though.

free, i.e., $P^\delta_{x/y} = P^\delta_{x/z}/P^\delta_{y/z}$. The corresponding "outer" spot prices exhibited by the pool to traders are

$$P = \left(P_{x/z} = \frac{\delta_x}{\delta_z} P^\delta_{x/z}, P_{y/z} = \frac{\delta_y}{\delta_z} P^\delta_{y/z}\right).$$

In this exposition, we will be mostly working in rate-scaled space, i.e., with the rate-scaled prices P^δ and p^δ.

Due to its structure and parameters, the pool can only attain a bounded set of P^δ vectors; we call these *feasible pool price vectors*. This set is *not* simply the square $[\alpha, 1/\alpha] \times [\alpha, 1/\alpha]$ but through the interaction of assets, it looks like Figure 1. Specifically, the set of feasible pool price vectors is the intersection of three conditions, labeled "$x \geq 0$", "$y \geq 0$", and "$z \geq 0$" in the figure (see below for the reasons for these labels).

For every vector of true (rate-scaled) prices, there is a unique *equilibrium pool price vector* $P^\delta(p^\delta)$ where no arbitrage exists, and it can be computed by projecting p^δ onto the boundary of the feasible region using a certain algorithm (Klages-Mundt and Schuldenzucker, 2023, Section 5.3).[2] In the previous reference, rate scaling is not considered, but arbitrage-freeness is not affected by rate scaling.[3] The pool is *out of range* if the current (rate-scaled) prices p^δ are not feasible, or equivalently $P^\delta(p^\delta) \neq p^\delta$.

Based on the reserve balances in the pool, the pool also exhibits an *actual* current vector of spot prices $P^\delta(t^\delta)$. The pool exposes an arbitrage opportunity iff $P^\delta(t^\delta) \neq P^\delta(p^\delta)$. In the following, we will assume that this is not the case (i.e., $P^\delta(t^\delta) = P^\delta(p^\delta)$), following the usual argument that any such opportunity would quickly be taken by arbitrageurs.[4]

3. Rate provider update

Assume now that we find ourselves in a situation where the pool is in equilibrium with the external market but out of range, i.e., $P^\delta(t^\delta) = P^\delta(p^\delta) \neq p^\delta$. In this situation, we have for at least one of the three asset pairs that no trading is possible in either direction at the current global market price. Our goal is now to update the rates δ_x and δ_y to new rates δ'_x and δ'_y such that the following two conditions hold:

1. **Arbitrage-freeness.** Updating the rates does not introduce an arbitrage opportunity.
2. **Efficiency.** The pool is in range after the update.

We show in the remainder of this paper that these two conditions can be achieved by chosing δ' in such a way as to move the rate-scaled market prices to the current equilibrium prices, i.e., to establish

$$p^{\delta'} = P^\delta(p^\delta).$$

This is achieved by choosing

[2] Note that the δ value in $P^\delta(...)$ does not refer to any specific δ but is merely notation to indicate that we refer to rate-scaled pool prices. The function $P^\delta(...)$ does not depend on the scaling rates δ.

[3] The *size* of an arbitrage opportunity, if there is one, is affected by rate scaling, though, up to a factor of $\max_{(i,j)\in\{x,y,z\}} \delta_i/\delta_j$.

[4] In the presence of swap fees, small price divergences (up to the fee) are not exploitable by arbitrageurs and therefore the the equality will only hold approximately. This is ignored here.

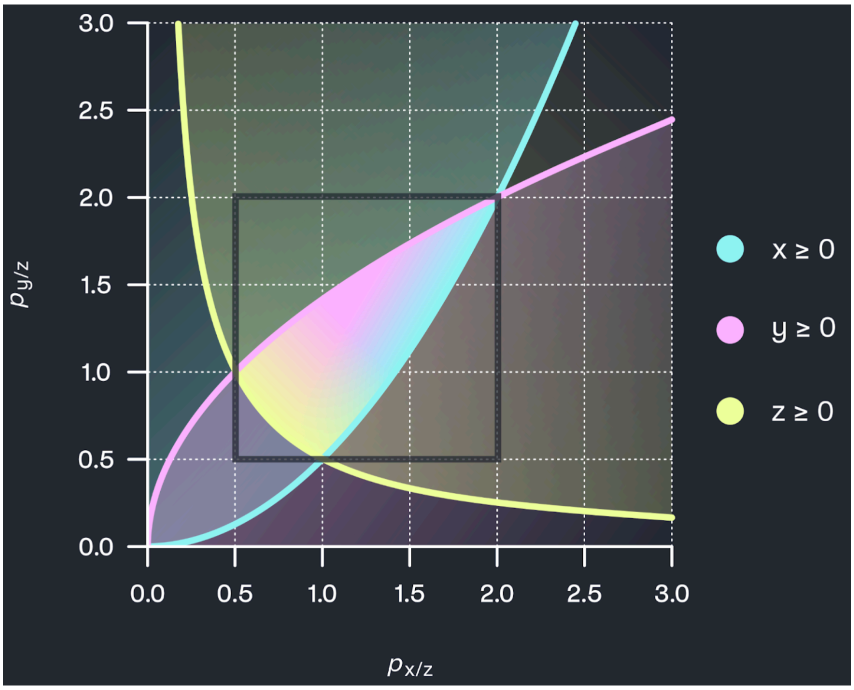

Figure 1: Space of feasible pool prices in the 3CLP. This is the intersection of three sets given by the condition that all implied balances need to be non-negative. Note that the axes should be labeled $P^\delta_{x/z}$ and $P^\delta_{y/z}$ instead of $p_{x/z}$ and $p_{y/z}$, respectively, in our notation here. The black square is $[\alpha, 1/\alpha] \times [\alpha, 1/\alpha]$. Note that the lower corner (α, α) of the square is not feasible, as are many other points in the square.

$$\begin{aligned}
\delta'_x &:= \delta_z \cdot \frac{p_{x/z}}{P^\delta_{x/z}(p^\delta)} = \delta_x \cdot \frac{p^\delta_{x/z}}{P^\delta_{x/z}(p^\delta)} \\
\delta'_y &:= \delta_z \cdot \frac{p_{y/z}}{P^\delta_{y/z}(p^\delta)} = \delta_y \cdot \frac{p^\delta_{y/z}}{P^\delta_{y/z}(p^\delta)} \\
\delta'_z &:= \delta_z.
\end{aligned}$$

Observe that, if p^δ is already feasible (i.e., $P^\delta(p^\delta) = p^\delta$), then the terms cancel out to yield $\delta'_x = \delta_x$ and $\delta'_y = \delta_y$, i.e., there is no update, as expected. Otherwise, there will be some change. Figure 2 illustrates some further example projections from p^δ to $p^{\delta'} = P^\delta(p^\delta)$. Essentially, we always project onto the closest point at the boundary along the direction(s) of the condition(s) that is/are not satisfied. Note that this projection is the same as equilibrium calculation in Klages-Mundt and Schuldenzucker (2023, Section 5.3).

The remainder of this document is to show that this update is indeed arbitrage-free and efficient. To do this, we first show that the update does not affect rate-scaled prices.

4. No change in rate-scaled prices

The updated rates δ' need to be carefully chosen because the actual (not rate-scaled) balances of the pool do (of course) not change due to our rate provider update while the rate-scaled pool balances (which determine the prices exhibited by the “inner” pool curve) do, from t^δ to $t^{\delta'}$. Because of this, for general δ', an update could introduce an

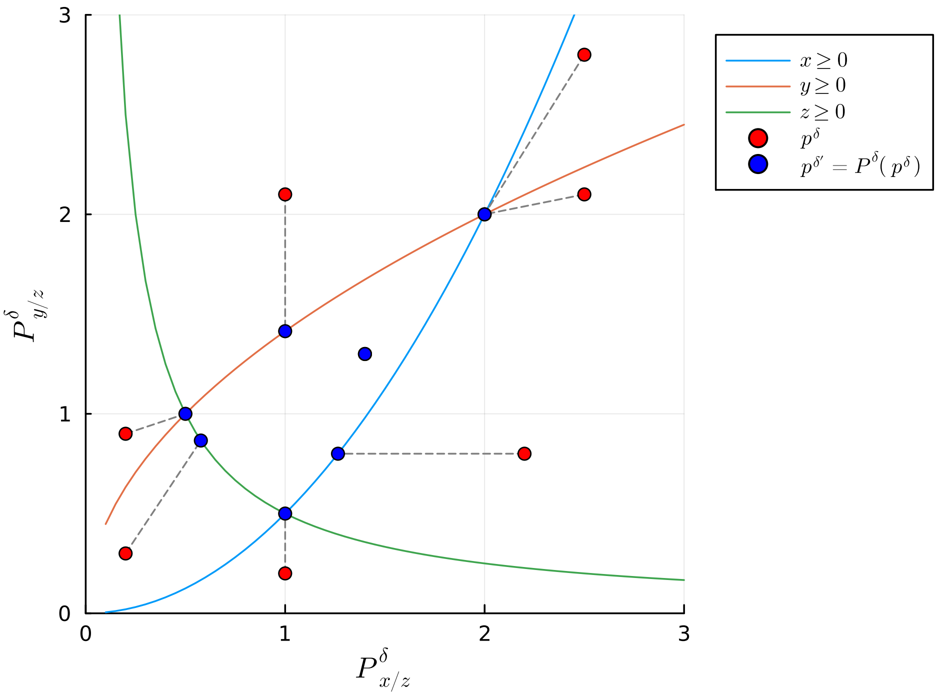

Figure 2: Some example projections of the rate-scaled price vector p^{δ} under the original rates to the new rates vector $p^{\delta'}$, which coincides with price equilibrium computation. The set labeled "$x \geq 0$" is also denoted T_x below, and analogously for y and z.

arbitrage opportunity. We need to show that this is not the case for our choice of δ' and to do this, we need to consider the interaction between balances and prices in greater detail.

Let $P^{\delta}(t^{\delta})$ be the rate-scaled pool prices exhibited at rate-scaled balances t^{δ}. See Klages-Mundt and Schuldenzucker (2022, Section 3) for the corresponding formulas and note that there is a 1:1 correspondence between P^{δ} vectors and pairs (t^{δ}, L), where L is the pool invariant, and $t^{\delta}(P^{\delta}, L)$ scales linearly in L.

Our main lemma towards arbitrage-freeness and efficiency is to observe that the update from δ to δ' does not change the rate-scaled pool prices.

Lemma 1:

$$P^{\delta}\left(t^{\delta'}\right) = P^{\delta}(t^{\delta})$$

Proof: Denote the interior of a set A as A° and the boundary ∂A. Klages-Mundt and Schuldenzucker (2023, Appendix A.2) represent the set of feasible pool price vectors as the intersection of three sets $T_x \cap T_y \cap T_z$ such that we have $P^{\delta}(p^{\delta}) \in T_x \cap T_y \cap T_z$ and

$$(*) \quad P^{\delta}(t^{\delta}) \in \partial T_x \Leftrightarrow x = 0$$

and likewise for the other assets.

We perform case distinction to show that $t^{\delta'} = \lambda \cdot t^{\delta}$ for some scalar $\lambda > 0$. It is easy to see that this implies the statement of the lemma.

First, if $P^\delta(p^\delta) \in T_x^\circ \cap T_y^\circ \cap T_z^\circ$, then this implies that p^δ is already feasible, so by the above discussion, $\delta' = \delta$ and there is no change.

Second, if $P^\delta(p^\delta) \in \partial T_i \cap \partial T_j \cap T_j^\circ$ for some permutation $\{i, j, k\} = \{x, y, z\}$, then by ($\star$), two of the asset balances x, y, z are zero and then $t^{\delta'}$ and t^δ are trivially related by a scalar.

Third, if $P^\delta(p^\delta) \in \partial T_i \cap T_j^\circ \cap T_k^\circ$, we perform case distinction over i.

If $i = x$, then we have $P^\delta_{y/z}(p^\delta) = p^\delta_{y/z}$ (Klages-Mundt and Schuldenzucker, 2023, Algorithm 1 and Theorem 1) and therefore $\delta'_y = \delta_y$, and also $x = 0$ by ($\star$). This implies $t^{\delta'} = t^\delta$ and in particular, they are related by the scalar 1.

If $i = y$, the analogous statement applies.

If $i = z$, then $\frac{P^\delta_{x/z}(p^\delta)}{p^\delta_{x/z}} = \frac{P^\delta_{y/z}(p^\delta)}{p^\delta_{y/z}}$ (see the algorithm and theorem again) and therefore $\frac{\delta'_x}{\delta_x} = \frac{\delta'_y}{\delta_y}$, and again by ($\star$) we have $z = 0$. This again implies the scaling property.

This concludes our case distinction because $\partial T_x \cap \partial T_y \cap \partial T_z = \emptyset$. ■

5. Proving the two core properties

We can now prove the two core properties.

Theorem 1: *The update from δ to δ' is arbitrage-free and efficient.*

Proof: We have

$$P^\delta\left(t^{\delta'}\right) = P^\delta(t^\delta) = P^\delta(p^\delta) = p^{\delta'} = P^\delta\left(p^{\delta'}\right)$$

where

- the first equality is Lemma 1,
- the second is by assumption since we assume that the pool was in equilibrium before the update,
- the third is by choice of δ',
- and the fourth is because $p^{\delta'}$ is feasible because it is chosen equal to the feasible price vector $P^\delta(p^\delta)$.

This proves arbitrage-freeness because the (rate-scaled) pool prices post-update $P^\delta\left(t^{\delta'}\right)$ are equal to the equilibrium (rate-scaled) pool prices $P^\delta\left(p^{\delta'}\right)$, i.e., the pool is in equilibrium post-update, i.e., there is no arbitrage opportunity. Efficiency follows because these prices are also equal to the (rate-scaled) market prices $p^{\delta'}$.

■

6. Remarks

The two-asset case can be seen as a trivial special case of the 3-asset construction laid out here: if we only have two assets, we would usually have an updatable rate provider on only one asset (say, x) and we would update it such that the pool moves just to the edge of its price range (which is a one-dimensional interval for two assets). This can be viewed as computing the equilibrium price (which is either the current price if the pool is in range or one of the two edges of the price range if it's not) and then scaling the price in the same way as we did here.

Our construction should also generalize to other multi-asset pool types beyond the 3CLP, and to more than 3 assets. The only place where we used knowledge of the details of the pool curve was Lemma 1. We feel that the properties that are used in the proof should hold for any sufficiently well-formed AMM, though. Specifically, any AMM should satisfy that certain balances are 0 at the boundary of its feasible price region and the direction of the projection should correspond to a set of equilibrium conditions that are violated. A variant of the proof of Klages-Mundt and Schuldenzucker (2023, Lemma 1 in Appendix A.2) might be helpful to derive a general form for this. This might also yield a general algorithm for equilibrium computation.

Our update always preserves a state where some of the pool balances are 0, meaning that even post-update, these assets can be sold to the pool (at market spot prices) but not bought. This form of an update therefore relies on later price movements to move the market price fully back into the feasible range. This is the best we can achieve while keeping the real (non-scaled) pool balances the same. An extension could assume that during the update procedure, we can trade at (or close to) market prices at some other market venue; then, we would essentially be able to update the the rates arbitrarily to move the pool to a better liquidity region beyond the edge of the price range. In practice, liquidity in other venues and the costs of trading would have to be carefully traded off against improved availability of the pool.

Bibliography

Klages-Mundt, A. and Schuldenzucker, S. (2022) "Concentrated Liquidity Pools with 2 or 3 Assets via Constant Products with Virtual Reserves". Available at: https://docs.gyro.finance/pools/technical-documents.

Klages-Mundt, A. and Schuldenzucker, S. (2023) "Design of the Gyroscope Consolidated Price Feed and Circuit Breaker System". Available at: https://docs.gyro.finance/gyd/technical-documents#consolidated-price-feed.