Concentrated Liquidity Pools with 2 or 3 Assets via Constant Products with Virtual Reserves

Ariah Klages-Mundt* Steffen Schuldenzucker†

August 21, 2022

Abstract

We describe a construction for *concentrated liquidity pools* with 2 or 3 assets (short *2-CLP* and *3-CLP*). The pools are constructed as constant-product market makers with virtual reserves. The possible pricing range of these AMMs is restricted to an interval; this is done by adding carefully chosen offsets to the "real" reserve amounts. Concentrated liquidity improves capital efficiency because the AMM only has to hold capital for prices that are actually expected to occur. While the 2-asset variant is known as a simplified version of the mechanism used in Uniswap v3, our 3-asset construction is new. Our method further generalizes to any number of assets.

1 Preliminaries

We consider CFMMs $f(x)$ in reserve space with any number of assets $(x_1, \dots, x_n)$. We single out x_n as the *numéraire*, i.e., prices will be denoted w.r.t. x_n. The *price* of asset i is

$$p_i := -\frac{\mathrm{d}x_n}{\mathrm{d}x_i},$$

where the derivative is taken along the curve where $f(x)$ stays constant. The price of i measures, in a marginal sense, by how much the numéraire x_n has to increase if x_i decreases by an (infinitesimal) unit and all other reserves stay the same.[1] Note that $p_n = 1$. We also write (x, y) or (x, y, z) instead of the variables $(x_1, \dots, x_n)$ and p_x and p_y accordingly; in this context, we also write $t := (x, y)$ or $t := (x, y, z)$ for the whole reserve state.

*Researcher at Superluminal Labs, a software development company working in the Gyroscope ecosystem; in a separate capacity, a PhD student at Cornell University. `ariah@gyro.finance`

†Researcher at Superluminal Labs. `steffen@gyro.finance`

[1] See also Klages-Mundt and Schuldenzucker (2021) for a more in-depth discussion.

2 2-asset variant

The 2-asset constant-product market maker with virtual reserves was previously described in the Uniswap v3 whitepaper (Adams et al., 2021). Even though we are not the first to write down this mechanism, it may be instructive to perform the construction from first principles. In contrast to Uniswap, the 2-CLP described in this document only specifies the mechanism with *one* pair of price bounds; in Uniswap, this would be called a single *position*. Uniswap supports a large number of (user-defined) positions and performs additional work to ensure this structure can be manipulated efficiently. Some other features that have been present since Uniswap v2 are also not described here; these include flash swaps, sync and skim functions.

Given are price bounds $[\alpha, \beta]$ such that $\alpha < \beta$.[2] We want to find a and b such that the following holds. Consider the curve of points (x, y) such that

$$(x + a)(y + b) = L^2, \tag{1}$$

where L is an arbitrary constant. Then a and b should be chosen such that, if $x = 0$, then $p_x = \beta$ and if $y = 0$, then $p_x = \alpha$. In particular, $p_x \in [\alpha, \beta]$ for all points (x, y) and these bounds are tight. In general, a and b can be arbitrary expressions; however, we will see that we can choose them as functions of only L (more specifically, as constant multiples of L). We will see that L is a useful *liquidity invariant*. The mechanism will track L, in addition to other constants.

2.1 Choosing Virtual Reserves

The following lemma describes how the offsets a and b need to be chosen. In the following, for the sake of clarity, we always write x' and y' in any context that is applied to the virtual reserves and we write x and y for the real reserves. Concretely, $x' = x + a$ and $y' = y + b$.

Lemma 1. *The above conditions are satisfied iff*

$$\begin{aligned} a &= L/\sqrt{\beta} \\ b &= L \cdot \sqrt{\alpha}. \end{aligned}$$

Furthermore,

$$\begin{aligned} x = 0 \Leftrightarrow \qquad & y = y^+ := L \cdot (\sqrt{\beta} - \sqrt{\alpha}) \\ y = 0 \Leftrightarrow \qquad & x = x^+ := L \cdot (1/\sqrt{\alpha} - 1/\sqrt{\beta}) \end{aligned}$$

[2] More standard notation would be to call these bounds $[p, q]$, but we will use these letters in a different context below. To avoid any confusion, we rename the variables here. Note that, in many applications, we will have $\alpha < 1 < \beta$, but this is not required for the pool to work.

Proof. Let $f(x, y) := xy$ be the invariant function of the CPMM without virtual reserves and let $g(t) = f(t + (a, b))$ be the CPMM *with* virtual reserves, i.e., the 2-CLP. Observe that $g(t) = L^2$ iff (1) holds. Write $t' := t + (a, b) =: (x', y')$ for the virtual reserves. The price of asset x' in the mechanism f is $p_x^f(t') = y'/x'$. Since a and b are (required to be) constant in L, they don't change along the curve of (x, y) pairs satisfying (1). From this it follows (see, e.g., Klages-Mundt and Schuldenzucker (2021)) that for the 2-CLP, the price of x is simply

$$p_x(t) = p_x^f(t') = \frac{y'}{x'} = \frac{y + b}{x + a}. \tag{2}$$

This is of course equivalent to $y + b = p_x(x + a)$ and since (1) should be satisfied at the same time, we have

$$p_x(x + a)^2 = L^2$$

at all points on the curve. Specifically at $x = 0$ we need $p_x = \beta$ and thus

$$\beta a^2 = L^2$$

and thus $a = L/\sqrt{\beta}$. Likewise, (2) and (1) imply

$$(y + b)^2/p_x = L^2$$

and for $y = 0$ we need $p_x = \alpha$ and thus

$$b^2/\alpha = L^2,$$

which implies $b = L \cdot \sqrt{\alpha}$. The formulas for x^+ and y^+ immediately follow from the computed values for a and b and the invariant equation (1). $\square$

For some applications (such as optimal order routing), we need the derivative of the price. This is easily computed:

Lemma 2. *The derivative of the price of x dependent on changes in x and y, respectively, along the trading curve, is*

$$\frac{\mathrm{d}p_x}{\mathrm{d}x} = -2 \cdot \frac{y'}{x'^2}$$
$$\frac{\mathrm{d}p_x}{\mathrm{d}y} = 2 \cdot \frac{1}{x'}$$

Let $p_{x \text{ per } y} = 1/p_x$ is the price of y denoted in units of x. Then

$$\frac{\mathrm{d}p_{x \text{ per } y}}{\mathrm{d}x} = 2 \cdot \frac{1}{y'}$$
$$\frac{\mathrm{d}p_{x \text{ per } y}}{\mathrm{d}y} = -2 \cdot \frac{x'}{y'^2}.$$

Proof. We have (see above) $p_x = \frac{y'}{x'} = \frac{L^2}{x'^2}$, where the second equality is by the invariant equation (1). Since L is kept constant along the trading curve by definition,

$$\frac{\mathrm{d}p_x}{\mathrm{d}x} = \frac{\mathrm{d}p_x}{\mathrm{d}x'} = -2 \cdot \frac{L^2}{x'^3} = -2 \cdot \frac{y'}{x'^2},$$

where the first identity is because $x' = x + a$ and a is constant along the trading curve and the last equality is by the invariant. For the second identity, note that also $p_x = \frac{y'}{x'} = \frac{y'^2}{L^2}$ and thus

$$\frac{\mathrm{d}p_x}{\mathrm{d}y} = \frac{\mathrm{d}p_x}{\mathrm{d}y'} = 2 \cdot \frac{y'}{L^2} = 2 \cdot \frac{1}{x'}.$$

The identities for $p_{x \text{ per } y}$ follow by symmetry. □

2.2 Standard Operations

We now discuss how to perform standard operations in the 2-CLP pool.

2.2.1 Initialization from real reserves

We now show how to initialize a pool based on the amount of reserves x and y. For a more gas-efficient way that is based on p_x and L, see below.

Initialization from real reserves requires computing the liquidity invariant L based on x and y. This operation is, perhaps surprisingly, not completely trivial, e.g., in terms of gas. Already for the CPMM *without* virtual reserves, it is easy to compute $L^2 = x \cdot y$, but to compute L from this, we need to take a square root, which is a somewhat significant operation in terms of gas. One might simply store the square $k := L^2$ in this case to avoid the square root; however, this would make liquidity updates more expensive (see Section 2.2.3 for the 2-CLP).

With virtual reserves, an additional complication arises: if we replace the above values for a and b in (1), then both sides of the equation depend on L. This is normal: virtual reserves allow us to use a larger L than we otherwise would, making the curve flatter and adding liquidity in terms of a lower price impact within the specified bounds. The following proposition shows how to solve this equation. This operation is still somewhat gas-intensive because we still need to compute a square root.

Proposition 1. *For any $0 \le \alpha < \beta$ and any $x, y \ge 0$, there exists a unique $L \ge 0$ such that (1) holds when the values for a and b are chosen like in Lemma 1. Specifically,*

$$L = (1 - \sqrt{\alpha}/\sqrt{\beta})^{-1} \cdot \left[\frac{1}{2} \cdot (y/\sqrt{\beta} + x\sqrt{\alpha}) + \sqrt{\frac{1}{4} \cdot (y/\sqrt{\beta} + x\sqrt{\alpha})^2 + (1 - \sqrt{\alpha}/\sqrt{\beta})xy} \right].$$

Proof. (1) is equivalent to

$$
\begin{aligned}
0 &= L^2 - (x+a)(y+b) \\
&= L^2 - (x + L/\sqrt{\beta})(y + L \cdot \sqrt{\alpha}) \\
&= \left(1 - \sqrt{\alpha/\beta}\right) L^2 - \left(y/\sqrt{\beta} + x\sqrt{\alpha}\right) L - xy.
\end{aligned}
$$

The coefficient of L^2 is positive since $\alpha < \beta$ and the coefficient of L and the constant intercept are obviously non-positive. This implies that this equation has a unique non-negative solution for L and it is equal to the specified formula (to see this, consider, for instance, the quadratic formula). □

2.2.2 Initialization from price

When a new pool is initialized, we can sometimes assume that a price p_x is given exogenously (e.g., from another market). We may now want to initialize a pool based on one of two further numbers: the value of the liquidity invariant L or the portfolio value

$$V := p_x x + y.$$

Note that the liquidity invariant L is *not* meaningful in the context of AMMs of other shapes or even to the 3-CLP; in contrast, the portfolio value V is universally meaningful and can be used to compare the 2-CLP to other AMMs. We first describe how to initialize x and y based on p_x and L.

Remark 1. If there are no virtual reserves, we have $p_x x = \frac{y}{x} \cdot x = y$ and thus $V = 2y$. This is a well-known property of the CPMM without virtual reserves: the two assets always contribute with equal values to the reserve. When there are virtual reserves, this statement need no longer true. More in detail, one can show using Lemma 1 and simple algebra that $p_x x = \frac{y'}{x'} \cdot x = y$ iff $p_x = \sqrt{\alpha\beta}$. One recovers the equations for the CPMM from the 2-CLP in the limit for $\alpha \to 0$ and $\beta \to \infty$.

Lemma 3. *In a 2-CLP pool with price bounds $[\alpha, \beta]$, price $p_x \in [\alpha, \beta]$, and liquidity invariant L we have*

$$
\begin{aligned}
x &= L \cdot (1/\sqrt{p_x} - 1/\sqrt{\beta}) \\
y &= L \cdot (\sqrt{p_x} - \sqrt{\alpha})
\end{aligned}
$$

The 2-CLP obviously cannot offer any prices outside its price range. If the exogenous price is $\hat{p}_x < \alpha$, then we have to choose $x = x^+$ and $y = 0$ and if $\hat{p}_x > \beta$ we choose $x = 0$ and $y = y^+$. This is the unique choice that does not create an arbitrage opportunity with the external market.

Proof of Lemma (3). Like in the proof of Lemma 1 we have

$$p_x(x+a)^2 = L^2$$
$$(y+b)^2/p_x = L^2.$$

The statement now follows immediately from the values of a and b computed in Lemma 1. $\square$

We now show how to compute the portfolio value V based on the price p_x and the liquidity invariant L.

Proposition 2. *In a 2-CLP with price bounds $[\alpha, \beta]$, price $p_x \in [\alpha, \beta]$, and liquidity invariant L, we have*

$$V = L \cdot \left[2\sqrt{p_x} - p_x/\sqrt{\beta} - \sqrt{\alpha}\right].$$

Proof. Follows immediately by plugging the values for x and y from Lemma 3 into the definition of V and simplifying. $\square$

We can also apply this proposition in reverse to compute L from V and p_x. This yields the following:

Lemma 4. *In a 2-CLP with price bounds $[\alpha, \beta]$, price $p_x \in [\alpha, \beta]$, and portfolio value V, we have*

$$L = \frac{1}{2\sqrt{p_x} - p_x/\sqrt{\beta} - \sqrt{\alpha}} V$$
$$x = \frac{1/\sqrt{p_x} - 1/\sqrt{\beta}}{2\sqrt{p_x} - p_x/\sqrt{\beta} - \sqrt{\alpha}} V$$
$$y = \frac{\sqrt{p_x} - \sqrt{\alpha}}{2\sqrt{p_x} - p_x/\sqrt{\beta} - \sqrt{\alpha}} V$$

Proof. Follows from Proposition 2 and Lemma 3. $\square$

2.2.3 Liquidity Update

A common operation is to add or remove liquidity, changing the liquidity invariant L while keeping the price p_x the same.

Proposition 3. *Assume (x, y, L) satisfy (1) at price p_x and let $\Delta L \in [-L, \infty)$. Then $(x + \Delta x, y + \Delta y, L + \Delta L)$ satisfy (1) at price p_x iff*

$$\Delta x = \Delta L \cdot \left(1/\sqrt{p_x} - 1/\sqrt{\beta}\right)$$
$$\Delta y = \Delta L \cdot \left(\sqrt{p_x} - \sqrt{\alpha}\right)$$

If some external price $\hat{p}_x \notin [\alpha, \beta]$ is applied, then one of x or y is 0 and only the other reserve component should be updated.

Proof of the proposition. This follows immediately from Lemma 3. □

An LP who wants to add/withdraw a certain amount of *portfolio value* could use Lemma 4 to compute the amounts Δx and Δy she would have to provide/withdraw to do this without changing the price. This is a simple operation because L is linear in V (if p_x remains unchanged) and x and y are linear in L, so the whole operation is linear.

When the real reserves x, y are known (as they usually are in practical situations where one wants to update liquidity), we receive a simpler formula for the liquidity update:

Corollary 1. *Assume (x, y, L) satisfy (1) at price p_x and let $\Delta L \in [-L, \infty)$. Then $(x + \Delta x, y + \Delta y, L + \Delta L)$ satisfy (1) at price p_x iff*

$$\frac{\Delta x}{x} = \frac{\Delta y}{y} = \frac{\Delta L}{L}.$$

Proof. This follows immediately by combining Proposition 3 with Lemma 3. The factor that only depends on p_x cancels out. □

Note that a liquidity provider who wants to add or remove liquidity cannot simply choose the composition of the assets she adds/removes, as that would change the price. LPs who want to do this need to combine their liquidity update with an appropriate trade to receive the right composition of assets.

2.2.4 Trade (Swap) Execution

Trade execution is simple in the 2-CLP. The following proposition provides a general way to specify this: a trader can provide or request an amount of x or of y.

Proposition 4. *Assume that (x, y, L) satisfy (1). Then $(x + \Delta x, y + \Delta y, L)$ satisfy (1) iff*

$$\Delta y = \frac{L^2}{x' + \Delta x} - y' = \frac{y' \Delta x}{x' + \Delta x}$$

and, equivalently,

$$\Delta x = \frac{L^2}{y' + \Delta y} - x' = \frac{x' \Delta y}{y' + \Delta y}.$$

Such values exist (in such a way that none of the new reserves $x + \Delta x$ and $y + \Delta y$ are negative) iff $\Delta x \in [-x,\, x^+ - x]$ and $\Delta y \in [-y,\, y^+ - y]$, respectively, where x^+ and y^+ are like in Lemma 1.

Proof. Note that additive changes to the real reserves affect the virtual reserves equally, i.e., we have $(x + \Delta x)' = x + \Delta x + a = x' + \Delta x$ and likewise for y. The equations now immediately follow from (1) for the updated reserves,

$$(x' + \Delta x)(y' + \Delta y) = L^2.$$

The respective equalities follow from $L = x'y'$ and simple algebraic transformation. The bounds in the last sentence of the proposition follow immediately from Lemma 1. $\square$

One would use Proposition 4 to execute a trade as follows: Assume that a trader wants to exchange an amount of Δx of asset x for asset y (the other direction being symmetric). If $\Delta x > 0$, the trader wants to sell asset x to the mechanism for asset y and if $\Delta x < 0$, the trader wants to buy asset x for asset y. We first use the last sentence of the proposition to check if this is even possible. We then use the formula in the theorem to compute Δy from Δx. We receive (if $\Delta x > 0$) / pay out (if $\Delta x < 0$) the amount Δx from/to the user and receive/pay out (respectively) Δy to/from the user. This operation changes the instantaneous price is to the new value

$$p_{x,(x+\Delta x, y+\Delta y)} = \frac{y' + \Delta y}{x' + \Delta x}.$$

Note that we can replace L^2 in Proposition 4 by $x'y'$ when these quantities are known. This may be convenient from a technical point of view.

For some applications, like optimal order routing, we need to compute the price of the asset that leaves the pool priced in units of the asset that goes into the pool, and the derivative of that price. We can easily map this to the results we already have. In the following, we assume that asset y goes into the pool and x comes out, i.e., the trader buys x for y from the mechanism; the other case is of course symmetric.

Lemma 5. *Assume that (x, y, L) satisfy (1) and consider the functions*

$$\begin{aligned} y_{\mathit{in}} &: [0, x] \to [0, y^+ - y] \\ x_{\mathit{out}} &: [0, y^+ - y] \to [0, x] \end{aligned}$$

defined as follows. If $\Delta y_{\mathit{in}} \in [0, y^+ - y]$, then $x_{\mathit{out}}(\Delta y_{\mathit{in}}) := \Delta x_{\mathit{out}}$ such that $(x - \Delta x_{\mathit{out}}, y + \Delta y_{\mathit{in}}, L)$ satisfy (1). If $\Delta x_{\mathit{out}} \in [0, x]$, then $y_{\mathit{in}}(\Delta x_{\mathit{out}}) := \Delta y_{\mathit{in}}$ such that $(x - \Delta x_{\mathit{out}}, y + \Delta y_{\mathit{in}}, L)$ satisfy (1). Note that we consider Δx_{out} and Δy_{in} in absolute value here. Then the following hold:

1. *$x_{\mathit{out}} \circ y_{\mathit{in}} = \mathrm{id}$ and $y_{\mathit{in}} \circ x_{\mathit{out}} = \mathrm{id}$.*
2. *$p_x(x - x_{\mathit{out}}(\Delta y_{\mathit{in}}),\, y + \Delta y_{\mathit{in}}) = \frac{y' + \Delta y_{\mathit{in}}}{x' - x_{\mathit{out}}(\Delta y_{\mathit{in}})} = 1 \Big/ \frac{\mathrm{d}x_{\mathit{out}}(\Delta y_{\mathit{in}})}{\mathrm{d}\Delta y_{\mathit{in}}}$ and*
   $p_x(x - \Delta x_{\mathit{out}},\, y + y_{\mathit{in}}(\Delta x_{\mathit{out}})) = \frac{y' + y_{\mathit{in}}(\Delta x_{\mathit{out}})}{x' - \Delta x_{\mathit{out}}} = \frac{\mathrm{d}y_{\mathit{in}}(\Delta x_{\mathit{out}})}{\mathrm{d}\Delta x_{\mathit{out}}}$.
3. *$\frac{\mathrm{d}p_x(x - x_{\mathit{out}}(\Delta y_{\mathit{in}}),\, y + \Delta y_{\mathit{in}})}{\mathrm{d}\Delta y_{\mathit{in}}} = 2 \cdot \frac{1}{x' - x_{\mathit{out}}(\Delta y_{\mathit{in}})}$ and*
   $\frac{\mathrm{d}p_x(x - \Delta x_{\mathit{out}},\, y + y_{\mathit{in}}(\Delta x_{\mathit{out}}))}{\mathrm{d}\Delta x_{\mathit{out}}} = 2 \cdot \frac{y' + y_{\mathit{in}}(\Delta x_{\mathit{out}})}{(x' - \Delta x_{\mathit{out}})^2}$.

Proof. 1. follows immediately by definition and by uniqueness.

2. follows by choice of the functions x_{out} and y_{in}. The respective first equality is easy to see.

3. follows from part 2 and Lemma 2, noting that the simple derivative $\frac{\mathrm{d}p_x(x-x_{\text{out}}(\Delta y_{\text{in}}),\,y+\Delta y_{\text{in}})}{\mathrm{d}\Delta y_{\text{in}}}$ is equal to the derivative $\frac{\mathrm{d}p_x}{\mathrm{d}y}$ along the trading curve, evaluated at $(x - x_{\text{out}}(\Delta y_{\text{in}}),\ y + \Delta y_{\text{in}})$, since this expression parameterizes the trading curve in the direction of y. Likewise, for the second equation we consider $\frac{\mathrm{d}p_x}{\mathrm{d}x}$ and note that $(x - \Delta x_{\text{out}},\ y + y_{\text{in}}(\Delta y_{\text{in}}))$ parameterizes the trading curve in the direction of $-x$. Since x decreases along this parameterization, another factor (-1) enters into the derivative (which follows immediately from, e.g., the chain rule). □

2.3 Integrating Fees

2.3.1 Swaps with Fees

Percentage fees are computed relative to the amount that flows *into* the contract.[3] Specifically, let γ be the percentage fee. Then a fee of $\gamma \max(\Delta x, \Delta y)$ is paid by the trader as follows: assume WLOG that the trader provides Δx; the case where she provides Δy is analogous. If $\Delta x > 0$, retain a fee of $\gamma\Delta x$, replace Δx by $(1-\gamma)\Delta x$, and execute the remainder of the trade as normal. If $\Delta x < 0$, first execute the trade as normal to receive an amount $\Delta y > 0$; ask the trader for an amount $1/(1-\gamma)\Delta y$, retain a fee of $\gamma/(1-\gamma)\Delta y$ of this and swap the remaining Δy into the contract.[4]Observe that no changes to the mathematical foundations are necessary since we simply retain part of the assets put into the swap.

We can describe the mechanism with fees in another way: let $\Delta x^+ := \max(0, \Delta x)$ and $\Delta x^- := \max(0, -\Delta x)$, and observe that $\Delta x = \Delta x^+ - \Delta x^-$ and at most one of $\Delta x^+, \Delta x^- \neq 0$; likewise for y. It is easy to see that the mechanism described in this paragraph chooses Δx or Δy given the other one such that

$$(x + (1-\gamma)\Delta x^+ - \Delta x^-) \cdot (y + (1-\gamma)\Delta y^+ - \Delta y^-) = L^2. \tag{3}$$

This was previously described by Angeris and Chitra (2020).

An additional design decision needs to be made now regarding what should be done with the fee of size $\gamma \max(\Delta x^+, \Delta y^+)$. There are two options: it can either simply be stored on a separate account, to be collected by the LP at their own discretion (*non-compounding fees*) or it can be automatically re-invested into the pool (*compounding fees*). We argue that compounding fees are desirable for two reasons: first, they improve capital efficiency for the fees generated, allowing them to serve LPs instead of sitting idle. Second, as we will see, they allow the fees to be stored *implicitly* in the real reserves, without keeping a fee account for each LP, reducing gas costs.

Remark 2 (Non-compounding fees). If we were to implement non-compounding fees, we

[3] See also Angeris and Chitra (2020) for some theoretical background for this section.

[4] This is the unique way of specifying fees where (1) the trader can specify if they want to send or receive money and (2) the fees are the same when they send X or receive Y and when they send Y or receive X, respectively.

would execute a trade with fees normally, then add the fees into the pool and update L based on the new reserves. When there are several LPs, fees accrue individually to a separate *fee balance* $f_i = (f_{i,x}, f_{i,y})$ that is stored for each LP i. Specifically, when a swap occurs, f_i increases by $L_i/L \cdot f$, where $f := \gamma(\Delta x^+, \Delta y^+)$ is the total fee in both assets. Note that one of f_x or f_y is 0. Uniswap v3 implements non-compounding fees. The main motivation for this is the potentially large amount of user-defined price bounds (Adams et al., 2021). In our case, we do not face this problem.

For some applications, like optimal order routing, we need to consider the "price" of an asset including fees, and we also need to consider its derivative. The following lemma shows how to calculate this. This generalizes Lemma 5.

Lemma 6. *Assume that (x, y, L) satisfy (1) and assume that there are swap fees of size γ. Let $\bar\gamma := 1 - \gamma$. Consider the functions*

$$\begin{aligned} y_{\mathit{in}} &: [0, x] \to [0, y^+ - y] \\ x_{\mathit{out}} &: [0, y^+ - y] \to [0, x] \end{aligned}$$

defined as follows. If $\Delta y_{\mathit{in}} \in [0, y^+ - y]$, then $x_{\mathit{out}}(\Delta y_{\mathit{in}}) := \Delta x_{\mathit{out}}$ such that $(x - \Delta x_{\mathit{out}}, y + \bar\gamma \Delta y_{\mathit{in}}, L)$ satisfy (1). If $\Delta x_{\mathit{out}} \in [0, x]$, then $y_{\mathit{in}}(\Delta x_{\mathit{out}}) := \Delta y_{\mathit{in}}$ such that $(x - \Delta x_{\mathit{out}}, y + \bar\gamma \Delta y_{\mathit{in}}, L)$ satisfy (1). Let $p_{x,\gamma,\mathit{in}}(\Delta y_{\mathit{in}}) := \frac{1}{\bar\gamma} p_x(x - x_{\mathit{out}}(\Delta y_{\mathit{in}}),\, y + \Delta y_{\mathit{in}})$ and $p_{x,\gamma,\mathit{out}}(\Delta x_{\mathit{out}}) := \frac{1}{\bar\gamma} p_x(x - \Delta x_{\mathit{out}},\, y + y_{\mathit{in}}(\Delta x_{\mathit{out}}))$. Then the following hold:

1. *$x_{\mathit{out}} \circ y_{\mathit{in}} = \mathrm{id}$ and $y_{\mathit{in}} \circ x_{\mathit{out}} = \mathrm{id}$. Furthermore, $p_{x,\gamma,\mathit{in}} \circ y_{\mathit{in}} = p_{x,\gamma,\mathit{out}}$ and $p_{x,\gamma,\mathit{out}} \circ x_{\mathit{out}} = p_{x,\gamma,\mathit{in}}$.*
2. *$p_{x,\gamma,\mathit{in}}(\Delta y_{\mathit{in}}) = \frac{1}{\bar\gamma} \frac{y' + \bar\gamma \Delta y_{\mathit{in}}}{x' - x_{\mathit{out}}(\Delta y_{\mathit{in}})} = 1 \Big/ \frac{\mathrm{d} x_{\mathit{out}}(\Delta y_{\mathit{in}})}{\mathrm{d} \Delta y_{\mathit{in}}}$ and*
   $p_{x,\gamma,\mathit{out}}(\Delta x_{\mathit{out}}) = \frac{1}{\bar\gamma} \frac{y' + \bar\gamma y_{\mathit{in}}(\Delta x_{\mathit{out}})}{x' - \Delta x_{\mathit{out}}} = \frac{\mathrm{d} y_{\mathit{in}}(\Delta x_{\mathit{out}})}{\mathrm{d} \Delta x_{\mathit{out}}}$.
3. *$\frac{\mathrm{d} p_{x,\gamma,\mathit{in}}(\Delta y_{\mathit{in}})}{\mathrm{d} \Delta y_{\mathit{in}}} = 2 \cdot \frac{1}{x' - x_{\mathit{out}}(\Delta y_{\mathit{in}})}$ and*
   $\frac{\mathrm{d} p_{x,\gamma,\mathit{out}}(\Delta x_{\mathit{out}})}{\mathrm{d} \Delta x_{\mathit{out}}} = \frac{1}{\bar\gamma} \cdot 2 \cdot \frac{y' + \bar\gamma y_{\mathit{in}}(\Delta x_{\mathit{out}})}{(x' - \Delta x_{\mathit{out}})^2}$.

Note that computing the virtual reserves x' and y' requires knowledge of the invariant L. Note further that, since we are only interested in an individual trader, it is not relevant for this computation if fees compound or not.

Proof. 1. follows immediately by definition.

2. The respective first equality follows immediately by definition. Towards the second equality in the first line, we can calculate explicitly

$$\begin{aligned} x_{\text{out}}(\Delta y_{\text{in}}) &= x' - \frac{L^2}{y' + \bar\gamma \Delta y_{\text{in}}} \\ y_{\text{in}}(\Delta x_{\text{out}}) &= \frac{1}{\bar\gamma} \left(\frac{L^2}{x' - \Delta x_{\text{out}}} - y' \right) \end{aligned}$$

and thus

$$\frac{\mathrm{d}x_{\mathsf{out}}(\Delta y_{\mathsf{in}})}{\mathrm{d}\Delta y_{\mathsf{in}}} = \bar{\gamma}\frac{L^2}{(y' + \bar{\gamma}\Delta y_{\mathsf{in}})^2} = \bar{\gamma}\frac{x' - x_{\mathsf{out}}(\Delta y_{\mathsf{in}})}{y' + \bar{\gamma}\Delta y_{\mathsf{in}}} = \bar{\gamma}/p_x(x - x_{\mathsf{out}}(\Delta y_{\mathsf{in}}), y + \bar{\gamma}\Delta y_{\mathsf{in}})$$
$$\frac{\mathrm{d}y_{\mathsf{in}}(\Delta x_{\mathsf{out}})}{\mathrm{d}\Delta x_{\mathsf{out}}} = \frac{1}{\bar{\gamma}}\frac{L^2}{(x' - \Delta x_{\mathsf{out}})^2} = \frac{1}{\bar{\gamma}}\frac{y' + \bar{\gamma}y_{\mathsf{in}}(\Delta x_{\mathsf{out}})}{x' - \Delta x_{\mathsf{out}}} = \frac{1}{\bar{\gamma}}p_x(x - \Delta x_{\mathsf{out}}, y + \bar{\gamma}y_{\mathsf{in}}(\Delta x_{\mathsf{out}})).$$

3. From part 2, we receive

$$\frac{\mathrm{d}p_{x,\gamma,\mathsf{in}}(\Delta y_{\mathsf{in}})}{\mathrm{d}\Delta y_{\mathsf{in}}} = \frac{\mathrm{d}}{\mathrm{d}\Delta y_{\mathsf{in}}}\frac{1}{\bar{\gamma}}\frac{(y' + \bar{\gamma}\Delta y_{\mathsf{in}})^2}{L^2} = 2\frac{y' + \bar{\gamma}\Delta y_{\mathsf{in}}}{L^2} = 2\frac{1}{x' - x_{\mathsf{out}}(\Delta y_{\mathsf{in}})}$$
$$\frac{\mathrm{d}p_{x,\gamma,\mathsf{out}}(\Delta x_{\mathsf{out}})}{\mathrm{d}\Delta x_{\mathsf{out}}} = \frac{\mathrm{d}}{\mathrm{d}\Delta x_{\mathsf{out}}}\frac{1}{\bar{\gamma}}\frac{L^2}{(x' - \Delta x_{\mathsf{out}})^2} = 2\frac{1}{\bar{\gamma}}\frac{L^2}{(x' - \Delta x_{\mathsf{out}})^3} = 2\frac{1}{\bar{\gamma}}\frac{y' + \bar{\gamma}y_{\mathsf{in}}(\Delta x_{\mathsf{out}})}{(x' - \Delta x_{\mathsf{out}})^2} \square$$

2.3.2 Slippage and Normalized Liquidity

Slippage is defined as the relative deviation of the effective (non-marginal) price of a trade to the marginal price before the trade. Following Lemma 6, we consider the effective price including fees and assume that asset y is swapped into and asset x is swapped out of the pool. Then the *marginal price including fees* is

$$p_{x,\gamma} := p_{x,\gamma,\mathsf{in}}(0) = p_{x,\gamma,\mathsf{out}}(0) = \frac{1}{\bar{\gamma}}p_x,$$

evaluated at the reserve state (x, y) before the swap. The *effective price including fees* is the amount that went into the pool (before fees) relative to the amount that left the pool, i.e.,

$$P_{x,\gamma} := \frac{\Delta y_{\mathsf{in}}}{\Delta x_{\mathsf{out}}},$$

when $\Delta y_{\mathsf{in}}, \Delta x_{\mathsf{out}}$ are chosen such that the invariant holds. Note that $P_{x,\gamma}$ depends on Δy_{in} and Δx_{out}, but $p_{x,\gamma}$ does not. *Slippage* is now the quotient of the two, normalized to 0, i.e.,

$$S := \frac{P_{x,\gamma}}{p_{x,\gamma}} - 1$$

as a function of either Δy_{in} or Δx_{out}. Call these two variants $S_{\mathsf{in}}(\Delta y_{\mathsf{in}})$ and $S_{\mathsf{out}}(\Delta x_{\mathsf{out}})$. For some applications (like optimal order routing), we are also interested in the linearizations (i.e., derivatives) of these functions at 0. The following lemma shows how to compute these values.

Lemma 7.

1. *$S = \frac{\bar{\gamma}\Delta y_{in}}{y'} = \frac{\Delta x_{out}}{x' - \Delta x_{out}}$.*
2. *$\frac{\mathrm{d}S_{in}(\Delta y_{in})}{\mathrm{d}\Delta y_{in}}(0) = \frac{\bar{\gamma}}{y'}$ and $\frac{\mathrm{d}S_{out}(\Delta x_{out})}{\mathrm{d}\Delta x_{out}}(0) = \frac{1}{x'}$.*

Proof. 1.: Observe that

$$\Delta x_{\text{out}} = x' - \frac{L^2}{y' + \bar{\gamma}\Delta y_{\text{in}}} = x' - \frac{x'y'}{y' + \bar{\gamma}\Delta y_{\text{in}}} = \frac{x'(y' + \bar{\gamma}\Delta y_{\text{in}}) - x'y'}{y' + \bar{\gamma}\Delta y_{\text{in}}} = \frac{\bar{\gamma}x'\Delta y_{\text{in}}}{y' + \bar{\gamma}\Delta y_{\text{in}}}$$

$$\Delta y_{\text{in}} = \frac{1}{\bar{\gamma}}\left(\frac{L^2}{x' - \Delta x_{\text{out}}} - y'\right) = \frac{1}{\bar{\gamma}}\frac{x'y' - y'(x' - \Delta x_{\text{out}})}{x' - \Delta x_{\text{out}}} = \frac{1}{\bar{\gamma}}\frac{y'\Delta x_{\text{out}}}{x' - \Delta x_{\text{out}}}.$$

Therefore,

$$\frac{\Delta y_{\text{in}}}{\Delta x_{\text{out}}} = \Delta y_{\text{in}}\frac{y' + \bar{\gamma}\Delta y_{\text{in}}}{\bar{\gamma}x'\Delta y_{\text{in}}} = \frac{1}{\bar{\gamma}}\frac{y'}{x'} + \frac{\Delta y_{\text{in}}}{x'} = p_{x,\gamma} + \frac{\Delta y_{\text{in}}}{x'}$$

$$\text{and also} \quad \frac{\Delta y_{\text{in}}}{\Delta x_{\text{out}}} = \frac{1}{\Delta x_{\text{out}}}\frac{1}{\bar{\gamma}}\frac{y'\Delta x_{\text{out}}}{x' - \Delta x_{\text{out}}} = \frac{1}{\bar{\gamma}}\frac{y'}{x' - \Delta x_{\text{out}}} = p_{x,\gamma}\frac{x'}{x' - \Delta x_{\text{out}}}.$$

Now $S = \frac{1}{p_{x,\gamma}}\frac{\Delta y_{\text{in}}}{\Delta x_{\text{out}}} - 1$ and thus

$$S = \frac{p_{x,\gamma}}{p_{x,\gamma}} + \frac{1}{p_{x,\gamma}}\frac{\Delta y_{\text{in}}}{x'} - 1 = \bar{\gamma}\frac{\Delta y_{\text{in}}}{y'}$$

$$\text{and also} \quad S = \frac{x'}{x' - \Delta x_{\text{out}}} - 1 = \frac{\Delta x_{\text{out}}}{x' - \Delta x_{\text{out}}}.$$

2.: For the first identity, part 1 implies that S is linear as a function of Δy_{in} and thus the result follows immediately. For the second identity, we calculate from part 1 that

$$\frac{\mathrm{d}S_{\text{out}}(\Delta x_{\text{out}})}{\mathrm{d}\Delta x_{\text{out}}} = \frac{x'}{(x' - \Delta x_{\text{out}})^2}$$

and evaluate this at $\Delta x_{\text{out}} = 0$. □

A standard notion of linearized slippage, introduced by Balancer,[5] corresponds to $\frac{\mathrm{d}S_{\text{in}}(\Delta y_{\text{in}})}{\mathrm{d}\Delta y_{\text{in}}}(0) = \frac{\bar{\gamma}}{y'}$. *Normalized liquidity* at a given point is defined as the inverse of standard linearized slippage, i.e.,

$$\text{Normalized Liquidity} = \frac{1}{\bar{\gamma}}y'.$$

Note that the values computed here are identical to those for Balancer's weighted-product pool[6] when all weights are equal to $1/2$ (i.e., the assets are weighted equally) and we in addition replace the real reserves y by the virtual reserves y'. Thus, virtual reserves merely introduce an offset in the calculation of slippage and normalized liquidity.

[5] See, e.g., `https://medium.com/balancer-protocol/calculating-value-impermanent-loss-and-slippage-for-balancer-pools-4371a21f1a86`

[6] See, e.g., the aforementioned blog post.

2.3.3 Compounding Fees and Keeping the Invariant Implicit

Compounding fees can be implemented in a particularly simple way.[7] Assume that the user provides $\Delta x > 0$, the other cases are analogous. Assume the current invariant is L. Compute Δy such that $(x + (1-\gamma)\Delta x,\, y + \Delta y,\, L)$ satisfy (1). The new reserves after the trade are $(x + \Delta x,\, y + \Delta y)$. This will *not* satisfy (1) with L, but with some $L' > L$. We can use 1 to compute this new L'. Liquidity providers collect their fees by removing liquidity via Proposition 3 or Corollary 1. It is not necessary to store the fees collected by any individual LP. Note that, in contrast to non-compounding fees, the liquidity "invariant" will grow over time as trades are executed. It is therefore likely more gas-efficient to not store L at all, but simply to recompute it whenever an operation (swap or liquidity adjustment) is performed.[8]

Remark 3 (Additional price impact due to compounding). One should be aware that compounding fees also slightly exacerbate price impact when trades tend to go in the same direction. To see why this happens, assume that a trader wants to sell x for y, so that $\Delta x > 0$. Fees charged on the trade amount to $\gamma \Delta x$ of asset x and the trade implies a price impact that depresses the price of x. When the fees are to be re-invested, since they only consist of asset x, this makes the reserve more unbalanced towards x and therefore further depresses the price of asset x. Note that this is undesirable for the next trade if they sell x, but desirable if they buy x. Overall, this effect appears to be rather mild and we believe that the added liquidity is usually worth the additional price impact.

2.3.4 Accounting for LP Shares

When fees do not compound, we can simply use the liquidity invariant L to account for LP shares: an LP can increase L by ΔL to receive an amount of LP shares equal to ΔL. They can redeem these shares to reduce L by (the current price-equivalent of) what they put in. However, when fees compound, L changes over time and so each trader's amount of LP shares would have to change over time as well. This is very inconvenient for LPs and also likely gas-inefficient when these amounts need to be updated after each trade.

Instead, a separate accounting structure for LP shares should be used that provides an additional layer of indirection before L.[9] The amount of LP tokens is stored in a

[7]This section was inspired by Balancer and transfers how Balancer's weighted pool collects fees to the 2-CLP. It turns out that the mechanism is relatively generic as long as L can be computed with not too much computational effort.

[8]Note that compounding fees imply higher computational effort because swaps and liquidity updates now require computation of a square root, whereas they could be done using elementary operations without compounding. However, optimized square root implementations, like the PRBMath, imply gas costs that are significantly below a store operation. Therefore, the savings in gas costs from saved store operations at least cancel out the additional gas costs due to computation. For the same reason, while one might store L after a swap to save a square root for a following liquidity update, this is likely not gas-efficient. It might be possible to further reduce gas costs by exploiting the knowledge that $(x + (1-\gamma)\Delta x, y + \Delta y, L)$ satisfy (1) and that γ is relatively small.

[9]This is again inspired by Balancer.

state variable S_i for each LP position i. The mechanism also keeps track of $S := \sum_i S_i$. A liquidity update by LP i increases L by ΔL (where $\Delta L \in [-L, \infty)$) iff it increases S_i by $S \cdot \Delta L/L$. Equivalently, to change S_i to $S_i + \Delta S$, the LP needs to change L to $L + \Delta L$, where $\Delta L = L \cdot \Delta S/S$. To do this, amounts of the assets x and y need to be provided / are redeemed according to Proposition 3 or Corollary 1. This operation obviously increases S to $S + \Delta S$. When the pool is initialized with initial reserves, an initial (arbitrary) value for S must be chosen. It seems natural to choose $S = L$ after initialization, though Balancer's weighted pool uses $2 \cdot L$ or, more generally, $n \cdot L$, where n is the number of tokens in the pool, in the interest of making token amounts more comparable across pools.[10]

The following result shows that the real reserves in the pool are split up among LPs according to the LP shares. This is a basic requirement for LP share accounting.

Proposition 5. *Let $x_{\text{LP}\,i}$ and $y_{\text{LP}\,i}$ be the maximal amounts of asset x and y, respectively, that LP i can redeem their LP shares fore. Then*

$$x_{\text{LP}\,i} = \frac{S_i}{S} \cdot x$$
$$y_{\text{LP}\,i} = \frac{S_i}{S} \cdot y.$$

Proof. The maximal amount of asset x that LP i can redeem for corresponds to $\Delta L = -L \cdot S_i/S$ and is thus equal to

$$x_{\text{LP}\,i} = L \cdot S_i/S \cdot \left(1/\sqrt{p_x} - 1/\sqrt{\beta}\right) = \frac{S_i}{S} \cdot L \left(1/\sqrt{p_x} - 1/\sqrt{\beta}\right) = \frac{S_i}{S} \cdot x.$$

Likewise for y. □

It is furthermore easy to see that adding liquidity and then removing it again is equivalent to no operation at all, so that fees only go to those LPs that were present when those fees accrued, and proportionally so.

In fact, the same linearity that drives the result of the previous proposition implies that the invariant L does not even have to be considered when new LP shares are minted or redeemed, as the following proposition shows.

Proposition 6. *An LP's liquidity update changes x by an absolute amount Δx, y by Δy, and S by ΔS iff*

$$\frac{\Delta x}{x} = \frac{\Delta y}{y} = \frac{\Delta S}{S}.$$

Proof. When a liquidity update is done following the rules above, then

$$\frac{\Delta S}{S} = \frac{\Delta L}{L} = \frac{\Delta x}{x} = \frac{\Delta y}{y},$$

[10] See `WeightedPool2Tokens.sol`, function `_onInitializePool` in the Balancer repo.

where the first equality is by definition and the others are by Corollary 1. The other direction follows by uniqueness. □

Remark 4. The simplified form of accounting for LP shares laid out in Proposition 6 is only possible as long as we can require that the two amounts Δx and Δy are provided in proportions defined by the protocol. As soon as LPs can also provide assets in an unbalanced way, the price changes during an LP update and the above technique does not work anymore.

2.3.5 Protocol Fees

In some infrastructures, like Balancer, pools need to pay *protocol fees*, which go to the creator of the pool infrastructure. The amounts received by LPs are reduced by the protocol fees.[11] Using the kind of accounting outlined here, it is easy to implement protocol fees. Consider a time where the outstanding amount of protocol fees are to be paid. Let L_0 be the value of the invariant the last time protocol fees were paid and let L be the value of the invariant now, and assume the protocol fees are equal to a factor δ. Then the protocol fees can be paid by reducing liquidity by an amount $\Delta L := \delta(L - L_0)$ and paying assets x and y proportionally to the protocol according to Proposition 3 or Corollary 1. The protocol thus acts as a "virtual LP." Note that none of the actual LP share amounts S_i or S need to be adjusted because their accounting is always relative to L. At the end of the process, the state variable L_0 needs to be updated to L to enable accounting for the next time protocol fees are paid. In Balancer's own implementation in the weighted pool, this process is executed before every liquidity update. It likely makes sense to keep it this way because LPs can usually afford the extra gas required, and also, doing this ensures that the pool always has sufficient assets to pay its protocol fees.

Paying Protocol Fees in LP Tokens. We can take the above idea one step further: instead of paying out the amounts corresponding to the protocol fees in the underlying assets, we can mint new LP tokens and pay the protocol fees directly in them. This has a number of advantages, most importantly gas fees because we save three token transfers. Another advantage is that protocol fees do not affect the invariant (instead, it affects the supply of the LP tokens), which simplifies the interaction between protocol fees and a liquidity update. Paying protocol fees in LP tokens is equivalent to first extracting them as assets, where then the protocol acts as an LP and puts those tokens back into the pool.

[11]This section is again inspired by Balancer's weighted pool. Note that Balancer's protocol fees are currently set to 0%, but may become positive in the future. Note also that the specification leaves significant leeway in how the protocol fees should be interpreted. For example, the weighted pool pays the protocol fees using a single asset in the pool (namely, the highest-weight one) while the implementation described here always pays them in a mix of both assets. This has the advantage that protocol fees do not affect the price.

Proposition 7. *Protocol fees of factor δ due to a change in the invariant from L_0 to $L > L_0$ correspond to an amount of new LP tokens equal to*

$$\Delta S = S \cdot \frac{\Delta L}{L - \Delta L},$$

where $\Delta L := \delta(L - L_0)$.

Proof. We use the aforementioned equivalence. Consider first a situation where protocol fees are paid in the underlying assets. Then, by definition, this reduces the invariant by ΔL to $L - \Delta L$. When the assets are then put back into the pool, this increases the invariant again from $L - \Delta L$ to L and thus (see above) generates an amount of LP tokens ΔS that satisfies

$$\frac{\Delta S}{S} = \frac{\Delta L}{L - \Delta L}.$$ □

Remark 5 (Generality of the approach). We were able to re-use Balancer's fee and LP share accounting approach essentially unmodified, which suggests a great deal of generality. It appears that this approach can always be followed when the following conditions hold:

1. A swap (without fees) can be performed with acceptable gas cost, even if the value of the invariant is not known.
2. When both assets change by the same factor, the price stays the same.
3. We can compute with acceptable gas cost a transformation of the liquidity invariant (here, L) that is proportional to the real reserves at any fixed price level.

The second condition is a form of scale invariance that is given in most AMMs. It is implied by Proposition 3. It is convenient but not necessary for our purposes to use the liquidity invariant L in our calculations. LP accounting can be done without explicitly mentioning L following Proposition 6. However, it is not clear how *protocol fees* would be described in such a framework; above, the invariant L provides a definite, one-dimensional way of measuring by how much the pool has grown the last time fees were paid. It appears that such a measure is necessary for well-defined protocol fees.

Remark 6 (Non-compounding protocol fees). An alternative interpretation of protocol fees is in a *non-compounding* way, where the protocol receives a share of δ of all fees (i.e., a share $\delta\gamma\Delta x$ when $\Delta x > 0$). These two definitions are *not* in general equivalent, not even when we compare them in terms of the portfolio value sent to the protocol. Letting protocol fees compound has the same advantages and disadvantages as for regular swap fees, and we believe that overall, the advantages outweigh the drawbacks.

2.4 Implementation

The mechanism (to be precise, it's compounding-fee variant, see above), should track the following variables over the course of its operation:

- The constants $\sqrt{\alpha}$ and $\sqrt{\beta}$. These can either be computed once when the contract is initialized or the price bounds are changed, or they can be computed off-chain, provided as input, and only verified on-chain. In any case, the computational effort is limited as these bounds will only be adjusted very rarely.
- The real reserves x and y, obviously.

For every swap and liquidity update, the value L should be recomputed using Proposition 1. Observe that the virtual reserves $x' = x + a = x + L/\sqrt{\beta}$ and y' can be easily computed from these values.

Given these values, Proposition 4 provides an easy and gas-efficient way to execute trades. Adjusting liquidity (Proposition 3) requires the value $\sqrt{p_x}$. However, the square root in this expression does *not* have to be computed explicitly but can be read from the other variables, as the following lemma shows:

Lemma 8. $\sqrt{p_x} = L/x'$

Proof. We have $x'y' = L^2$ and $p_x = y'/x'$. This implies $p_x = L^2/x'^2$, which implies the statement. □

The preceding statement is not helpful when newly initializing a 2-CLP at a certain price. In this case, $\sqrt{p_x}$ has to be computed or checked explicitly. Of course, this is trivial in the (common) special case when the initial price is 1.

3 3-asset variant

We now consider a 3-asset concentrated liquidity pool (*3-CLP*) based on a constant-product curve with virtual reserves. The real reserve is of form (x, y, z) and the invariant is

$$(x + a)(y + b)(z + c) = L^3. \tag{4}$$

Again, we will see that it is useful to specify the invariant as L^3 since many values will be linear in L. We assume and prove in the following that the offsets a, b, c are functions of L only. We use z as the numéraire. We first consider the general case, where arbitrary price bounds are provided. In the majority of this section, we will then consider a *symmetric* variant, where the price bounds must satisfy additional symmetry conditions. This will greatly simplify the involved calculations.

There are two relevant prices in the 3-CLP: $p_x = \frac{\mathrm{d}z}{\mathrm{d}x}$ and $p_y = \frac{\mathrm{d}z}{\mathrm{d}y}$. Note that the the of x denoted in y, and vice versa, are determined by p_x and p_y because $p_{x \text{ per } y} = \frac{\mathrm{d}x}{\mathrm{d}y} = p_y/p_x$ and $p_{y \text{ per } x} = 1/p_{x \text{ per } y} = p_x/p_y$ (Klages-Mundt and Schuldenzucker, 2021).

Prices in the 3-CLP are analogous to the 2-dimensional variant:

Lemma 9. *The prices in the 3-CLP are*

$$p_x = \frac{z'}{x'}$$
$$p_y = \frac{z'}{y'},$$

where $x' := x + a$*,* $y' := y + b$*, and* $z' := z + c$ *are the virtual reserves.*

Proof. We use the tools from Klages-Mundt and Schuldenzucker (2021).[12] Let $f(x, y, z) = xyz$. Then $\nabla f = (yz, xz, xy)$ and thus

$$p_x^f = \frac{yz}{xy} = \frac{z}{x}$$
$$p_y^f = \frac{xz}{xy} = \frac{z}{y}.$$

Since a, b, c are constants, the prices under $g := f \circ (+(a, b, c))$ are simply the prices under f shifted by the virtual reserve offsets. □

We can use this to fit the 3-CLP to price bounds. Note that the virtual reserve offsets only offer three degrees of freedom a, b, c (as functions of L). We will see that this implies that *only three* of the price bounds for p_x and p_y can be specified and the fourth is implied by them, as are the bounds for $p_{y \text{ per } x}$ (and thus also for $p_{x \text{ per } y}$).

Proposition 8. *Let* $\alpha_x < \beta_x$ *and* $\alpha_y < \beta_y$*. Then the 3-CLP attains prices* $p_x \in [\alpha_x, \beta_x]$*,* $p_y \in [\alpha_y, \beta_y]$*, and this is tight, iff*

$$a = L \cdot \alpha_x^{-1/3} \alpha_y^{1/3} \beta_x^{-1/3}$$
$$b = L \cdot \alpha_x^{2/3} \alpha_y^{-2/3} \beta_x^{-1/3}$$
$$c = L \cdot \alpha_x^{1/6} \alpha_y^{1/3} \beta_x^{1/6}$$

and

$$\beta_y = \frac{\alpha_y}{\alpha_x} \beta_x. \tag{5}$$

In this case, furthermore, $p_{y \text{ per } x} \in [\alpha_{y \text{ per } x}, \beta_{y \text{ per } x}]$ *tightly iff* $\alpha_{y \text{ per } x} = \alpha_x^{3/2} \alpha_y^{-1} \beta_x^{-1/2}$ *and* $\beta_{y \text{ per } x} = \alpha_x^{1/2} \alpha_y^{-1} \beta_x^{1/2}$*.*

[12] These results follow essentially by the implicit function theorem and the multi-dimensional chain rule.

Proof. We need to find values a, b, c such that

$$\begin{aligned} p_x &= \frac{z+c}{x+a} \in [\alpha_x, \beta_x] \\ p_y &= \frac{z+c}{y+b} \in [\alpha_y, \beta_y] \\ \text{across those } x, y, z \text{ where } &(x+a)(y+b)(z+c) = L^3 \\ \text{and } &x \geq 0,\ y \geq 0,\ z \geq 0. \end{aligned}$$

This is equivalent to

$$\begin{aligned} p_x =& \frac{z'}{x'} \in [\alpha_x, \beta_x] \\ p_y =& \frac{z'}{y'} \in [\alpha_y, \beta_y] \\ \text{across those } x', y', z' \text{ where } & x'y'z' = L^3 \\ \text{and } & x' \geq a,\ y' \geq b,\ z' \geq c. \end{aligned}$$

By transforming the invariant and replacing z' and x', respectively, into the formulas for p_x and p_y, we receive

$$\begin{aligned} p_x &= \frac{L^3}{x'^2 y'} = \frac{y' z'^2}{L^3} \\ p_y &= \frac{L^3}{x' y'^2} = \frac{x' z'^2}{L^3}. \end{aligned}$$

From these formulas it immediately follows that p_x is maximized when the denominator in the first fraction is minimized, i.e., for $x' = a$ and $y' = b$; thus, its maximum value is $\frac{L^3}{a^2 b}$. Repeating this for the other three fraction yields that the following equalities need to hold:

$$\begin{aligned} &\text{I} \quad \alpha_x = \frac{bc^2}{L^3} && \text{II} \quad \beta_x = \frac{L^3}{a^2 b} \\ &\text{III} \quad \alpha_y = \frac{ac^2}{L^3} && \text{IV} \quad \beta_y = \frac{L^3}{ab^2}. \end{aligned}$$

We can solve equations I–III for a, b, c as follows. By equation I and III we have

$$\frac{\alpha_x L^3}{b} = c^2 = \frac{\alpha_y L^3}{a} \tag{6}$$

and therefore

$$b = a \cdot \frac{\alpha_x}{\alpha_y}. \tag{7}$$

Plugging this into II yields

$$\begin{aligned}
&& \beta_x = \frac{L^3}{a^2 \cdot a \cdot \frac{\alpha_x}{\alpha_y}} = \frac{L^3}{a^3} \cdot \frac{\alpha_y}{\alpha_x} \\
\Leftrightarrow && a = L \cdot \alpha_x^{-1/3} \alpha_y^{1/3} \beta_x^{-1/3}.
\end{aligned}$$

Plugging this in turn into (7) yields

$$\begin{aligned}
b &= L \cdot \alpha_y^{1/3} \alpha_x^{-1/3} \beta_x^{-1/3} \cdot \frac{\alpha_x}{\alpha_y} \\
&= L \cdot \alpha_x^{2/3} \alpha_y^{-2/3} \beta_x^{-1/3}.
\end{aligned}$$

Finally, using (say) the second equality of (6) we receive

$$\begin{aligned}
c &= \left(\frac{\alpha_y L^3}{L \cdot \alpha_y^{1/3} \alpha_x^{-1/3} \beta_x^{-1/3}} \right)^{1/2} \\
&= L \cdot \alpha_x^{1/6} \alpha_y^{1/3} \beta_x^{1/6}.
\end{aligned}$$

Equation IV now yields

$$\begin{aligned}
\beta_y &= \frac{L^3}{L \cdot \alpha_x^{-1/3} \alpha_y^{1/3} \beta_x^{-1/3} \cdot \left(L \cdot \alpha_x^{2/3} \alpha_y^{-2/3} \beta_x^{-1/3} \right)^2} \\
&= \frac{1}{\alpha_x \alpha_y^{-1} \beta_x^{-1}} = \frac{\alpha_y}{\alpha_x} \beta_x.
\end{aligned}$$

For the bounds for $p_{y \text{ per } x}$, we apply the same methodology as above and replace the values for a, b, c. We have

$$p_{y \text{ per } x} = \frac{y'}{x'} = \frac{L^3}{x'^2 z'} = \frac{y'^2 z'}{L^3}.$$

Like above, this is minimized and maximized, respectively, at

$$\begin{aligned}
\alpha_{y \text{ per } x} &= L^{-3} \cdot b^2 c = L^{-3} \cdot \left(L \cdot \alpha_x^{2/3} \alpha_y^{-2/3} \beta_x^{-1/3} \right)^2 \cdot L \cdot \alpha_x^{1/6} \alpha_y^{1/3} \beta_x^{1/6} \\
&= \alpha_x^{3/2} \alpha_y^{-1} \beta_x^{-1/2} \\
\beta_{y \text{ per } x} &= L^3 \cdot \left(a^2 c \right)^{-1} = L^3 \cdot \left(L \cdot \alpha_x^{-1/3} \alpha_y^{1/3} \beta_x^{-1/3} \right)^{-2} \cdot \left(L \cdot \alpha_x^{1/6} \alpha_y^{1/3} \beta_x^{1/6} \right)^{-1} \\
&= \alpha_x^{1/2} \alpha_y^{-1} \beta_x^{1/2}.
\end{aligned}$$

□

The bounds in the above proposition are rather asymmetric. This is because we have made the (arbitrary) choice to calibrate the offsets to the three bounds $\alpha_x, \beta_x, \alpha_y$ rather than (say) any other set of three of the four bounds for x and y. Note that (5) allows us to infer any one of the four price bounds given the other three. Therefore, Proposition (8) allows computing the virtual reserve offsets based on (essentially) any parameterization of the four price bounds, or of any combination of the six bounds for $p_x, p_y, p_{y \text{ per } x}$.

While it may be surprising at first that we can only choose 3 out of 4 price bounds freely, there is a simple intuitive explanation for this: with 3 assets, we only have 3 virtual reserve offsets to choose. These 3 variables only offer 3 degrees of freedom, so we can only fit 3 price bounds. This phenomenon generalizes to any number of assets greater 2. Specifically, an "n-CLP" with n assets has n virtual reserve offsets, but $n-1$ assets other than the numéraire and thus $2 \cdot (n-1)$ price bounds one may want to calibrate to. And, of course $n < 2 \cdot (n-1)$ for all $n > 2$. For $n = 2$, the two are equal and the two price bounds can be chosen freely. In addition, the bounds for the prices p_{x_i} also determine the bounds for prices $p_{x_j \text{ per } x_i}$.[13]

The following corollary shows that we can always choose equal price bounds for p_x and p_y. However, to also have the price bounds for the third asset pair, $p_{y \text{ per } x}$, equal to those, we need an additional condition.

Corollary 2. *In the nomenclature of Proposition 8, if $\alpha_x = \alpha_y$, then also $\beta_x = \beta_y$. In this case, $\alpha_{y \text{ per } x} = (\alpha_x/\beta_x)^{1/2}$ and $\beta_{y \text{ per } x} = (\beta_x/\alpha_x)^{1/2} = 1/\alpha_{y \text{ per } x}$ and*

$$
\begin{aligned}
a = b &= L \cdot \beta_x^{-1/3} \\
c &= L \cdot \alpha_x^{1/2} \beta_x^{1/6}.
\end{aligned}
$$

Furthermore, we have $\alpha_{y \text{ per } x} = \alpha_x$ iff $\beta_{y \text{ per } x} = \beta_x$ iff $\beta_x = 1/\alpha_x$ and in this case $a = b = c = L \cdot \alpha_x^{1/3}$.

Proof. The first series of equalities follows immediately from Proposition 8. The last sentence follows immediately from the first part of the statement. □

We call the last case described in Corollary 2 the *symmetric* case for the 3-CLP and simply write α for $\alpha_x = \alpha_y = \alpha_{y \text{ per } x}$. Note how, if the bounds for p_x and p_y are equal, this automatically makes the price bounds for $p_{y \text{ per } x}$ "symmetric" in the sense that $\beta_{y \text{ per } x} = 1/\alpha_{y \text{ per } x}$. For the remainder of this document, we focus on the symmetric case. We require $\alpha < 1$ to ensure that the price bounds are non-degenerate.

Remark 7. In the symmetric case, the price range for *any* asset pair is equal to $[\alpha, 1/\alpha]$. More in detail, we have $p_{x \text{ per } z} = 1/p_x$ and this is tightly within $[\alpha/(1/\alpha), 1/\alpha] = [\alpha, 1/\alpha]$. Likewise for $p_{y \text{ per } z}$ and $p_{x \text{ per } y}$.

3.1 Standard Operations (Symmetric Case)

We describe the standard operations for a 3-CLP pool.

[13] Note that, e.g., $\alpha_{y \text{ per } x}$ is *not* usually equal to α_x/β_y even though $p_{y \text{ per } x} = p_x/p_y$. This is because the prices p_x and p_y cannot assume their extreme values independently of each other. Instead, the price bounds for $p_{y \text{ per } x}$ correspond to the formulas in Proposition 8.

3.1.1 Initialization from real reserves

Initializing a pool from real reserves is less simple than in the 2-asset case, even if the price bounds are symmetric. To do this, we need to compute L given only the real reserves x, y, z (and the price bounds). The associated equation (4) is now cubic in L, as opposed to quadratic like in the 2-asset case. It is easy to see that, in the symmetric case, (4) is equivalent to

$$(1-\alpha)L^3 - (x+y+z)\alpha^{2/3}L^2 - (xy+yz+xz)\alpha^{1/3}L - xyz = 0.$$

To initialize the pool, this equation needs to be solved for L. While one *could* solve this equation using the cubic formula, doing so requires taking two square roots and two cube roots, and one also potentially needs to deal with complex numbers. It seems more feasible to apply a numerical technique like Newton's method (noting that the derivatives of the left-hand side are easy to compute). Before deploying any such a method, one should first carefully study its numerical stability and running time in the context of this particular class of equations. Otherwise, exploitable errors could result. We discuss the design of such numerical methods in Appendix A.

As an alternative, a user could compute L off-chain and provide it along with x, y, z upon initialization. The contract would then simply check if the invariant holds. This would hurt composability for the operation of starting a new pool, but this seems to be a sensible trade-off. If there are no fees, none of the other operations require recomputation of L, as we will see in the following results. If there are fees, one can decide to compound fees and recompute the invariant on every swap.

3.1.2 Initialization from prices

Given prices p_x, p_y, we can initialize a pool in a somewhat analogous way to the 2-CLP. However, in contrast to the 2-CLP, the prices in the 3-CLP interact and therefore we cannot choose an arbitrary combination of prices even within the price bounds $[\alpha, 1/\alpha]$. Instead, another condition has to be considered.

Proposition 9. *Fix some $\alpha \in (0,1)$ and let $p_x, p_y > 0$. Consider the condition*

$$p_x p_y, \frac{p_x}{p_y^2}, \frac{p_y}{p_x^2} \geq \alpha. \tag{8}$$

(8) implies that $p_x, p_y, p_y/p_x \in [\alpha, 1/\alpha]$. If, in any symmetric 3-CLP, we have prices p_x and p_y and liquidity invariant L, then (8) holds and we furthermore have for the reserves that

$$
\begin{aligned}
x &= L \cdot \left(\sqrt[3]{p_y/p_x^2} - \alpha^{1/3}\right) &&= L \cdot \left(\sqrt[3]{p_x p_y}/p_x - \alpha^{1/3}\right) \\
y &= L \cdot \left(\sqrt[3]{p_x/p_y^2} - \alpha^{1/3}\right) &&= L \cdot \left(\sqrt[3]{p_x p_y}/p_y - \alpha^{1/3}\right) \\
z &= L \cdot \left(\sqrt[3]{p_x p_y} - \alpha^{1/3}\right) &&= L \cdot \left(\sqrt[3]{p_x p_y} - \alpha^{1/3}\right).
\end{aligned} \tag{9}
$$

Vice versa, if (8)*holds, then for any $L > 0$ there exists a unique valid reserve state (x, y, z) of a symmetric 3-CLP with invariant L and prices p_x, p_y and it is defined by the above equations.*

Proof. We first show that (8) is sufficient to imply that all prices are within the price bounds $[\alpha, 1/\alpha]$. To see this, note that

$$
\begin{aligned}
p_x^3 &= (p_x p_y)^2 \frac{p_x}{p_y^2} \\
p_y^3 &= (p_x p_y)^2 \frac{p_y}{p_x^2} \\
\left(\frac{p_y}{p_x}\right)^3 &= (p_x p_y) \left(\frac{p_y}{p_x^2}\right)^2
\end{aligned}
$$

are all $\geq \alpha^3$ by assumption and also

$$
\begin{aligned}
p_x^3 &= \left(\frac{p_y}{p_x^2}\right)^{-2} \left(\frac{p_x}{p_y^2}\right)^{-1} \\
p_y^3 &= \left(\frac{p_x}{p_y^2}\right)^{-2} \left(\frac{p_y}{p_x^2}\right)^{-1} \\
\left(\frac{p_y}{p_x}\right)^3 &= (p_x p_y)^{-1} \left(\frac{p_x}{p_y^2}\right)^{-2}
\end{aligned}
$$

are all $\leq 1/\alpha^3$.

Assume now that there is a reserve state with these prices and invariant L. We first prove the three equalities (9). We have $p_x = \frac{z'}{x'}$ and $p_y = \frac{z'}{y'}$ and thus $x' = z'/p_x$ and $y' = z'/p_y$ and, by the invariant,

$$
L^3 = x'y'z' = \frac{z'^3}{p_x p_y}.
$$

Rearranging yields $z' = L \cdot \sqrt[3]{p_x p_y}$. The two price equations above then yield $x' = z'/p_x = L \cdot \sqrt[3]{p_x p_y}/p_x$ and likewise $y' = L \cdot \sqrt[3]{p_x p_y}/p_y$. The formulas for x, y, z now follow from Corollary 2. From these equalities and because $x, y, z \geq 0$, it immediately follows that (8) must hold.

Towards the other direction, assume that (8) holds for some pair of prices p_x, p_y and let $L > 0$. Choose x, y, z according to (9). By assumption, $x, y, z \geq 0$. It is easy to see that (x, y, z, L) satisfy (4) and (via this) that the marginal prices in the pool are indeed p_x, p_y. Uniqueness follows by the first part of the proposition. □

Proposition 9 exhibits an interesting symmetry in the reserve values stemming from the symmetry in price bounds. Specifically, we have $x_i = L \cdot \left(\sqrt[n]{P}/p_i - \alpha\right)$ where $P = \prod_{i=1}^{n} p_i$ and n is the number of assets. It is easy to see that this holds for any number of assets n.

Using Proposition 9, computation of the portfolio value

$$V := p_x x + p_y y + z$$

is straightforward, just like in the 2-asset case, as the following proposition shows.

Proposition 10. *In a symmetric 3-CLP with price bounds α, prices $p_x, p_y \in [\alpha, 1/\alpha]$, and liquidity invariant L, we have*

$$V = L \cdot \left[3\sqrt[3]{p_x p_y} - (p_x + p_y + 1)\alpha^{1/3}\right]$$

Proof. Follows by plugging the values for x, y, z from Proposition 9 into the definition of the portfolio value and simplifying. □

It is easy to see that Proposition 10 generalizes to the n-asset case, where we receive $V = L \cdot \left[n\sqrt[n]{\prod_i p_i} - \sum_i p_i \alpha^{1/n}\right]$.

3.1.3 Liquidity Update

Updating liquidity is straightforward and analogous to the 2-asset case as well.

Proposition 11. *Assume (x, y, z, L) satisfy (4) at prices p_x, p_y and let $\Delta L \in [-L, \infty)$. Then $(x + \Delta x,\ y + \Delta y,\ z + \Delta z,\ L + \Delta L)$ satisfy (4) at prices p_x, p_y iff*

$$\begin{aligned}
\Delta x &= \Delta L \cdot \left(\sqrt[3]{p_x p_y}/p_x - \alpha^{1/3}\right) \\
\Delta y &= \Delta L \cdot \left(\sqrt[3]{p_x p_y}/p_y - \alpha^{1/3}\right) \\
\Delta z &= \Delta L \cdot \left(\sqrt[3]{p_x p_y} - \alpha^{1/3}\right).
\end{aligned}$$

Proof. Follows immediately from Proposition (9). □

Note that, like in the 2-asset case, the composition of assets that enter or leave the pool is determined by the prices and cannot be chosen by the trader. And just like in the 2-asset case, if the original values are known, we can greatly simplify the way they are calculated.

Corollary 3. *Assume (x, y, z, L) satisfy (4) at prices p_x, p_y and let $\Delta L \in [-L, \infty)$. Then $(x + \Delta x,\ y + \Delta y,\ z + \Delta z,\ L + \Delta L)$ satisfy (4) at prices p_x, p_y iff*

$$\frac{\Delta x}{x} = \frac{\Delta y}{y} = \frac{\Delta z}{z} = \frac{\Delta L}{L}.$$

Proof. This follows immediately by combining Proposition 11 with Lemma 9. The factor that only depends on p_x cancels out. □

3.1.4 Trade (Swap) Execution

Executing a trade is analogous to, but conceptually slightly more nuanced than the 2-asset case because traders may have different needs. Arguably the most common case is when a trader wants to swap one of the three assets for another one without providing or receiving any of the third asset. Other traders may also desire/provide a combination of two assets and provide/receive the third. All of these cases can be covered by solving the equation

$$(x' + \Delta x)(y' + \Delta y)(z' + \Delta z) = L^3,$$

which can be done easily and efficiently. A trader would provide two of the three values $\Delta x, \Delta y, \Delta z$ and we calculate the third one so that the invariant holds. In the case where the trader only wants to swap between a pair of assets, they would set the respective third Δx, Δy, or Δz to 0. We describe the swap for the case where the trader provides Δx and Δy. The other cases are analogous.

Proposition 12. *Assume that (x, y, z, L) satisfy (4) and let $\Delta x \geq -x$ and $\Delta y \geq -y$. Then $(x + \Delta x, y + \Delta y, z + \Delta z, L)$ satisfy (4) iff*

$$\Delta z = \frac{L^3}{(x' + \Delta x)(y' + \Delta y)} - z',$$

if this value is $\geq -z$. Otherwise, no such value exists.

Proof. Follows immediately from the invariant (4), just like Proposition (4). □

Note that, in contrast to Proposition (4), we cannot compute single bounds x^+, y^+, z^+ up to which one can trade. This is because, whether or not a trade is possible depends on the *combination* of values of the two provided Δ values. For example, we have $z = 0$ iff $z' = c$ iff

$$\begin{aligned} & x'y' = L^3/c \\ \Leftrightarrow \quad & (x + L \cdot \alpha^{1/3})(y + L \cdot \alpha^{1/3}) = L^2/\alpha^{1/3}. \end{aligned}$$

This is equivalent to saying that (x, y) lies on a 2-CLP with special virtual reserve offsets $L \cdot \alpha^{1/3}$ and liquidity invariant $L^2/\alpha^{1/6}$. We obviously cannot simplify this further to static values for x and y.

The most common kind of swap is a swap of one asset against another, where one of $\Delta x, \Delta y, \Delta z$ is 0. In this case, we can slightly simplify the involved operations. This may be convenient from a technical point of view because it reduces the number of variables involved in the calculation. We describe the case where asset x is swapped against asset z and asset y remains untouched. The other swap pairs are of course analogous.

Proposition 13. *Assume that* (x, y, z, L) *satisfy (4) and let* $\Delta x \geq -x$*. Then* $(x + \Delta x, y, z + \Delta z, L)$ *satisfy (4) iff*

$$\Delta z = \frac{x'z'}{x' + \Delta x} - z' = \frac{z'\Delta x}{x' + \Delta x},$$

if this value is $\geq -z$*. Otherwise, no such value exists.*

Proof. Follows immediately from Proposition 4. We have

$$\Delta z = \frac{L^3}{(x' + \Delta x)(y' + \Delta y)} - z' = \frac{x'y'z'}{(x' + \Delta x)y'} - z' = \frac{x'z'}{(x' + \Delta x)} - z'.$$

The second equality follows by $L^3 = x'y'z'$ and simple algebraic transformation. □

Asset y does not occur in the above calculation. Note, however, that the formula still depends on the invariant L because we need to compute the virtual assets x' and z'.

3.1.5 Fees

Fees can be included in much the same way as in the 2-CLP (see above): fees of factor γ are taken on any asset that goes into the pool. One difference is that now there can either be one or two assets going into the pool. Another is when fees should compound because re-initializing the liquidity invariant now requires solving a cubic equation (see Section 3.3). We argue that, despite the increased computational cost, doing so is still feasible and implies the advantages of compounding fees.

3.2 Optimal Arbitrage

Trading a pair of assets affects the price of the third. Assume, for instance, that a trader sells x for z. This decreases the amount of the reserve z (and thus of the virtual reserve z') while keeping y' constant and increasing x', which implies that the three prices $p_x = z'/x'$, $p_y = z'/y'$ and $p_{y \text{ per } x} = y'/x'$ all decrease. For a trader who does not care which of the two other assets they receive (such as an arbitrageur who believes that only asset x is mispriced), this is not optimal. Instead, assuming that the price of the third asset pair (p_y in our example) is in equilibrium, they have an incentive to trade in such a way as to *keep* it the same, as the following result shows. The intuition for this is that, when the price of the other asset changes, this creates an arbitrage opportunity; the trader misses out on value of the size of this arbitrage opportunity. We state the result for the case where a trader wants to exchange a fixed amount of x for a bundle of y and z. It is easy to see that the statement generalizes to any combination of assets.

For a trade $(\Delta x, \Delta y, \Delta z)$ let

$$\Delta V := -\left(p_x \Delta x + p_y \Delta y + \Delta z\right)$$

be the *value* of the trade. Importantly, prices are taken before the trade. This captures the assumption that prices are in equilibrium (at least the assets the trader is not interested in), but the trader does not care how they change after the trade. Note that ΔV captures the value of the trade from the perspective of the *trader*. In contrast, the *mechanism* incurs a value difference of $-\Delta V = p_x \Delta x + p_y \Delta y + \Delta z$.

Theorem 1. *Consider a 3-CLP with symmetric price bounds given by α such that (x, y, z, L) satisfy (4) and fix an amount $\Delta x \in [-x, \infty)$. Among all the pairs $(\Delta y, \Delta z)$ such that $(x + \Delta x, y + \Delta y, z + \Delta z, L)$ satisfy (4), ΔV is maximized iff $p_y^1 = p_y$, where p_y^1 is the price of asset y after the trade. This is the case iff*

$$\frac{\Delta z}{\Delta y} = p_y.$$

Proof. Note first that, with Δx fixed, maximizing ΔV is equivalent to maximizing $p_y \Delta y + \Delta z$. Consider now a trade where $p_y^1 \neq p_y$. Then there exists a valid (arbitrage) trade $(\Delta x^1 = 0, \Delta y^1, \Delta z^1)$ such $\Delta y^1, \Delta z^1 \neq 0$ starting at the state $(x + \Delta x, y + \Delta y, z + \Delta z)$ and the price p_y^2 after that trade is $p_y^2 = p_y$. Consider the combined trade $\Delta x^2 := \Delta x$, $\Delta y^2 := \Delta y + \Delta y^1$, $\Delta z^2 = \Delta z + \Delta z^1$. We show that the value ΔV^2 of this trade is higher than the value ΔV of $(\Delta x, \Delta y, \Delta z)$. Consider first the case where $p_y^1 < p_y = p_y^2$. Then $\Delta y^1 < 0 < \Delta z^1$ and therefore

$$\begin{aligned} \Delta V^2 - \Delta V = p_y \cdot (-\Delta y^1) - \Delta z^1 \geq 0 \\ \Leftrightarrow \quad p_y \geq \frac{\Delta z^1}{-\Delta y^1}. \end{aligned}$$

The last inequality holds because, for the trade $(\Delta y^1, \Delta z^1)$, the left-hand side of the inequality is the marginal price at the end of the trade and the right-hand side is the average price across the trade, and the trade consists of buying asset y. The other direction is analogous.

For the last statement of the theorem, note that

$$p_y^1 = \frac{z' + \Delta z}{y' + \Delta y} \stackrel{!}{=} \frac{z'}{y'} = p_y$$

iff $\frac{\Delta z}{\Delta y} = \frac{z'}{y'} = p_y$. This general result about fractions is easy to see. □

The statement that it is optimal to preserve the price of the third asset is actually much more general than stated and holds in any AMM that has sensible properties in some basic sense. This follows using the same proof as Theorem 1 because the proof only makes use of very general properties of the AMM (such as that selling an asset depresses the price). How a trader achieves this (i.e., the last part of the theorem) depends on the details of the AMM in question. Observe that, under the condition of the theorem, $p_y \Delta y = \Delta z$, i.e., the values of the two assets provided/received are the same. This echoes the property of the CPMM (without virtual reserves) whereby the three assets enter into the reserve with equal values.

3.3 Implementation (Symmetric Case)

The implementation should follow a similar structure as the 2-asset version and store the following items:

- The constant $\alpha^{1/3}$, either computed on the fly or provided and verified at construction of the pool.
- The liquidity invariant L (unless it is still more efficient to re-compute it each time).
- The real reserves x, y, z.

As before, the virtual reserves x', y', z' can be easily computed from these values. Similar to the 2-CLP, the root $\sqrt[3]{p_x p_y}$ can be efficiently computed from the tracked variables.

Lemma 10. $\sqrt[3]{p_x p_y} = \frac{L^2}{x'y'} = \frac{z'}{L}$.

Proof. By the invariant, we have $z' = L^3/(x'y')$ and thus

$$p_x p_y = \frac{z'}{x'} \cdot \frac{z'}{y'} = \frac{L^3}{x'^2 y'} \cdot \frac{L^3}{x' y'^2} = \frac{L^6}{x'^3 y'^3}.$$

This implies the first equality. The second follows by

$$\frac{L^2}{x'y'} = \frac{L^2 z'}{x'y'z'} = \frac{L^2 z'}{L^3} = \frac{z'}{L}. \qquad \square$$

Initializing the pool from a price requires actually computing the expression $\sqrt[3]{p_x p_y}$. Different methods, such as Newton's method, binomial expansion, or conversion to a pair of $\log/\exp$ expressions can be used towards this task. To initialize the pool from the real reserves x, y, z, a cubic equation needs to be solved (see Section 3.1.1).

A Numerical Methods to Compute the Invariant in the 3-Asset Case

Recall from Section 3.1.1 that computing the invariant from real reserves in the 3-asset case amounts to finding a non-negative solution the following equation:

$$(1-\alpha)L^3 - (x+y+z)\alpha^{2/3}L^2 - (xy+yz+xz)\alpha^{1/3}L - xyz = 0.$$

Assuming WLOG that $x, y, z > 0$ (otherwise, the problem becomes quadratic), this is obviously a cubic equation of form[14]

$$\begin{aligned} f(L) := aL^3 + bL^2 + cL + d = 0, \\ a > 0, \\ b, c, d < 0. \end{aligned}$$

We discuss the properties of such equations and two methods for solving them. Note that the coefficients a–d can be easily computed assuming that $\alpha^{1/3}$ is available. This could be precomputed or computed once at the beginning of the operation, whatever is more gas-efficient.

A.1 Shape of the function f

We first argue that the function f always exhibits the general shape depicted in Figure 1. First consider the derivatives of f:

$$\begin{aligned} f'(L) &= 3aL^2 + 2bL + c \\ f''(L) &= 6aL + 2b. \end{aligned}$$

Observe that

$$f''(L) = 0 \quad \Leftrightarrow \quad L = L_{\mathsf{M}} := \frac{-b}{3a}.$$

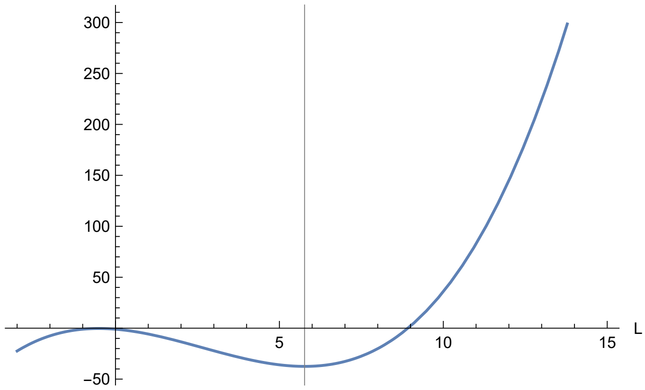
Since $b < 0 < a$, we have $L_{\mathsf{M}} > 0$ and $f''(L) < 0$ if $L < L_{\mathsf{M}}$ and $f''(L) > 0$ if $L > L_{\mathsf{M}}$. Furthermore, we have

$$\begin{aligned} f'(L) = 0 \quad \Leftrightarrow \quad L = L_{+,-} :=& \frac{-b}{3a} \pm \frac{\sqrt{b^2 - 3ac}}{3a} \\ =& L_{\mathsf{M}} \pm \sqrt{L_{\mathsf{M}}^2 - \frac{c}{3a}}. \end{aligned}$$

Since $c < 0$, this equation has two different solutions and $L_- < 0 < L_+$, and by the above discussion of f'', L_- is a local maximum and L_+ is a local minimum. Since $c < 0$, $L_- < 0 < L_+$. Since $f(0) < 0$ (and there are no other extrema but L_- and L_+),

[14] For this section, a, b, c do *not* refer to the virtual reserve offsets.

Figure 1 Cubic function f for $x = y = z = 1$ and $\alpha = 0.7$. The gray line marks the local minimum L_+.



$f(L) < 0 \,\forall L \in [0; L_+]$. Since there are no further roots of f', f is strictly increasing on $[L_+; \infty)$. Therefore, f has a unique root L_* in $[0, \infty)$ and $L_* > L_+$. Since $L_+ > L_{\mathsf{M}}$, f is also convex on $[L_+; \infty)$.

A.2 Finding the root L_* via Newton's method

The shape of the function f lends itself well to Newton's method. Consider a sequence of Newton iterations $(L_n)_n$. If $L_n > L_*$ then by monotonicity and convexity $L_n > L_{n+1} > L_*$. Thus, Newton's method converges whenever one of the steps L_n lies above L_*. Furthermore, if $L_n \in [L_+, L_*]$, then by monotonicity and convexity $L_{n+1} > L_*$. Therefore and extending the previous argument, Newton's method converges whenever it is started at some point in $L_0 \geq L_+$. To find such a point, it seems unavoidable to compute L_+ using a square root. Once L_+ is available, experiments have shown that $L_0 := 1.5 \cdot L_+$ yields a good first approximation of L_* unless α is very small (see Figure 2). Note that an extremely small value of α would likely not be chosen; here, the mechanism would be similar to a CPMM with three assets and *without* virtual reserves and one is likely better off using that simpler mechanism instead.

Experimentally, when accuracy is not limited by fixed-point numbers and other constraints, 5–7 Newton steps always achieve an error bound of 10^{-18}, measured in terms of the error in the values x, y, z when they are reconstructed using Proposition 9 using the current value for L. When only 18 decimal places are available for intermediate results during calculation, the ultimate error is larger. The accuracy of the method also depends on α, with higher α values leading to lower accuracy. Experimentally, we found that the method provides sufficiently accurate results for all realistic values of x, y, z, and α.

A.3 Convergence Speed, Stability, and Stopping Criterion

Realistic values are that x, y, z are in the millions to potentially billions and α is very close to 1, e.g., $\alpha = 0.9995$. For these values, the function f becomes very steep (see

Figure 2 Quotient L_*/L_+. The initial approximation $1.5 \cdot L_+$ lies at 1.5 on the Y axis. Observe that, unless α is very small, the initial approximation is reasonably close to L_*. In the most relevant case, where α is close to 1 (around 0.95, say), the initial approximation is very accurate. For $\alpha \to 1$, the initial approximation becomes exact in relative terms. Note that $\alpha = 1$ is not valid. The figure is for $x = y = z = 1$. Experiments have shown that the shape of this curve and accuracy of L_0 are very stable under different choices for the real reserves x, y, z, unless α is chosen very small.

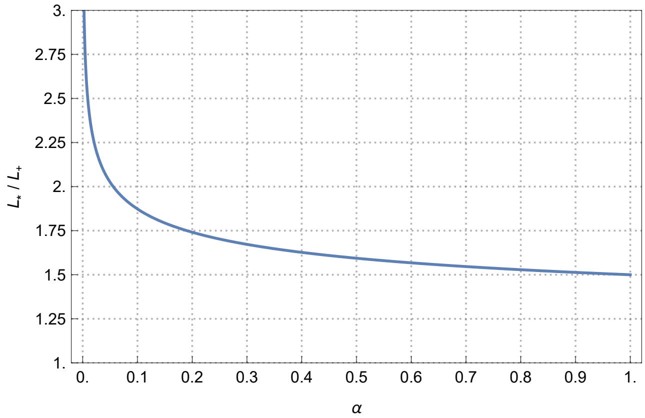


Figure 3 for a plot of f for large values of these parameters), implying that Newton's method converges rapidly. At the same time, $f(L)$ is very sensitive to errors in L. Note, however, that we fundamentally are not interested in the relative error $|f(L)|$ but rather in the absolute error $|L - L_*|$, which is why this sensitivity does not imply that we need to know L_* at a higher accuracy than would otherwise be desirable. In addition, while Newton's method converges from the right, for our application it is important to slightly *under*estimate L, since this will be more advantageous for the pool's reserves. Overall, this is why the following stopping criterion has proven effective: Iterate until (1) either the step size decreases by less than a factor 10 in a step or (2) the steps L_n increase rather than decrease.[15] Finally, to make sure we end up to the left of the invariant, make another step of the same size downwards, i.e., the final result is $L_n - |L_{n-1} - L_n|$.

A.4 Lower-order case

When one of x, y, z is 0, then $d = 0$ and thus $f(L) = 0$ if $L = 0$ or

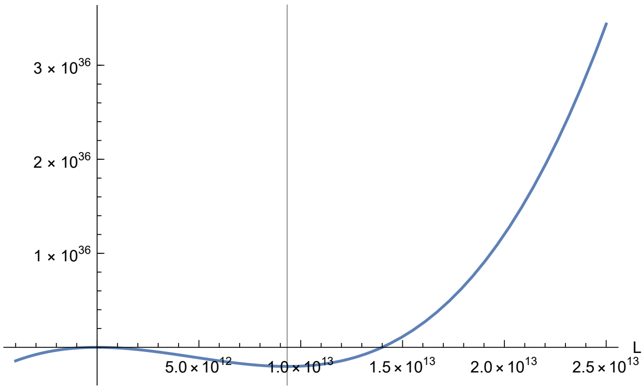
$$aL^2 + bL + c = 0.$$

This is of course the case iff

$$L = \frac{-b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a}.$$

[15]The second test checks for the influence of rounding errors. Note that this case happen mathematically (except in the first step), which tell us that the influence of rounding errors has become so large that further steps would not be informative.

Figure 3 Cubic function f for $x = y = z = (2, 3, 2) \cdot 10^9$ and $\alpha = 0.9995$. The gray line marks the local minimum L_+.



Since $x, y, z \geq 0$, we have $b, c \leq 0$ (and we always have $a > 0$), so that these terms are always defined and the "+" solution is non-negative, so that the maximal solution to $f(L) = 0$ is non-negative and is equal to

$$L_* = \frac{-b}{2a} + \frac{\sqrt{b^2 - 4ac}}{2a}.$$

If at least one of x, y, z is positive, then $L_* > 0$. If $x = y = z = 0$, then $L_* = 0$.

A.5 Future Work: Higher-order methods

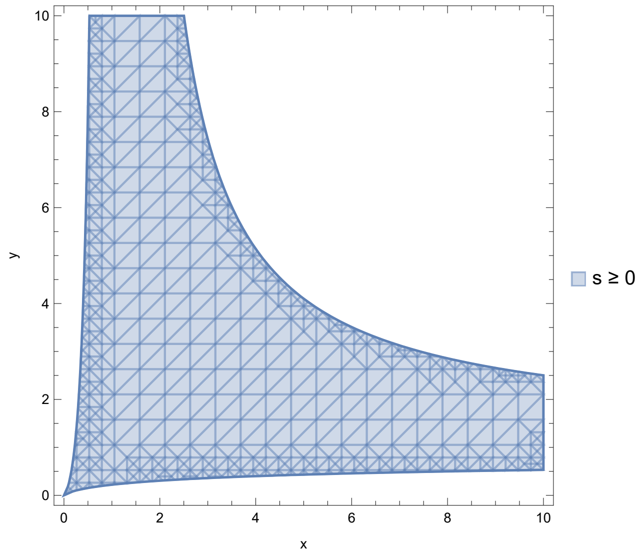
To further increase convergence speed, information about the second derivative could be included. This can be easily done using Halley's method, in which the tangent is replaced by the linear/linear Padé approximation as a local approximation of the function in question. Halley's method may provide better convergence, but it is not clear what its stability is compared to Newton's method. Anecdotally, Halley's method saves 1–2 iterations over Newton's.

A.6 Discussion: Finding the root L_* via the cubic formula?

Of course, a cubic equation can also be solved using the cubic formula. However, while this is generally possible, it appears infeasible using the methods currently available on-chain. As an exercise towards potential future work, we discuss how one might approach this. Let

$$\begin{aligned}
p &:= \frac{-b}{3a} \\
q &:= p^3 + \frac{bc - 3ad}{6a^2} \\
r &:= \frac{c}{3a} \\
s &:= q^2 + \left(r - p^2\right)^3.
\end{aligned}$$

Figure 4 Region of (x, y) pairs where $s \geq 0$. We have fixed $z = 1$ and $\alpha = 0.7$. Smaller α values lead to a larger region and larger α values lead to a slightly smaller region (but the region does not disappear for $\alpha \to 1$).



Then the solutions to the cubic equation are

$$\sqrt[3]{q + \sqrt{s}} + \sqrt[3]{q - \sqrt{s}} + p, \tag{10}$$

where all possible 3rd roots need to be considered to receive all possible solutions.[16] It is important to note that complex numbers can occur in intermediate steps of the calculation even if the end result is real. In particular, we cannot assume that s is non-negative. More in detail, observe that $p, q > 0$ and $r < 0$ by the signs of the coefficients. Thus, $(r - p^2)^3 < 0$ and s can be positive or negative.[17]

By the above discussion on the shape of f, the above term can take on exactly one non-negative value L_*. To find it, we only need to perform a limited number of operations on complex numbers. We distinguish two cases.

Case 1: First assume that $s \geq 0$ so that $\sqrt{s}$ is real. Since $(r - p^2)^3 < 0$, $\sqrt{s} < q$ and thus $q - \sqrt{s} > 0$. It follows that all radicands are non-negative and thus all roots are real. We need to compute one square root and two third roots to compute L_* via (10).

Case 2: Now assume that $s < 0$. Then $\sqrt{s}$ is purely imaginary (i.e., a multiple of i) and thus $q + \sqrt{s}$ and $q - \sqrt{s}$ are conjugate mixed complex numbers. Observe that $q + \sqrt{s} = q + i\sqrt{-s}$ is in the first quadrant of the Cartesian plane. Note that this $s < 0$ case is common and cannot be ignored in practice; to see this, consider Figure (4), which shows the "easy" region where $s \geq 0$ for an example choice of the other parameters.

To calculate the cube roots in (10), it seems that there is no other way than using

[16] In contrast, we do not need to consider the two possible square roots of s, because of symmetry. WLOG denote by $\sqrt{s}$ the positive (or positive-imaginary) root.

[17] This can not only happen in general, but also in our specific context: let $\alpha = 0.7$, $x = y = 1$, and $z = 0.1$. Then calculation shows that $s \approx -2.025$.

polar coordinates. Let $\varphi = \arctan(\sqrt{-s}/q)$ and observe that $|q + \sqrt{s}| = (q^2 - s)^{1/2}$ (this does *not* need to be calculated), so we have

$$q + \sqrt{s} = \left(q^2 - s\right)^{1/2} \cdot e^{i\varphi}$$
$$q - \sqrt{s} = \left(q^2 - s\right)^{1/2} \cdot e^{-i\varphi}.$$

Let now $t = (q^2 - s)^{1/6}$. Then we can consider the (specific) third roots

$$\sqrt[3]{q + \sqrt{s}} = t \cdot e^{i\varphi/3}$$
$$\sqrt[3]{q - \sqrt{s}} = t \cdot e^{-i\varphi/3}.$$

Since these are again complex conjugates of each other, their imaginary parts will cancel out and (10) is equal to

$$2\Re\left(\sqrt[3]{q + \sqrt{s}}\right) + p = 2t\cos(\varphi/3) + p.$$

Overall, we receive

$$L_* = 2 \cdot \left(q^2 - s\right)^{1/6} \cdot \cos\left(\frac{1}{3} \cdot \arctan\left(\frac{\sqrt{-s}}{q}\right)\right) + p.$$

Overall, in this case, we need to compute a square root, a sixth root, an arc tangent, and a cosine. While the case $s \geq 0$ case appears somewhat acceptable to compute on-chain, the need to deal with trigonometrics in the $s < 0$ case makes this approach unattractive for implementation on-chain.[18]

B Marginal Trading Curves and Capital Efficiency of the 3-CLP

We study the trading curves offered by the 3-CLP for two of the assets when we hold the third asset fixed. We call this the *marginal* trading curve. This will then allow us to compare the 2-CLP to the 3-CLP in terms of capital efficiency.

We first show that the marginal trading curve of a 3-CLP is equivalent to a 2-CLP with price bounds dependent on the third asset. The following theorem shows this when the third asset is y; the other cases are of course symmetric. We only consider the symmetric case. Recall that the offsets are $a = b = c = L\alpha^{1/3}$.

Theorem 2. *Consider a symmetric 3-CLP with price bound parameter α, assets x, y, z, and invariant L, and consider the curve of points (x_1, z_1) such that (x_1, y, z_1) satisfies*

[18] To the best of our knowledge, no on-chain implementation is currently available for the arc tangent.

Equation (4). This is the same curve as the trading curve of a 2-CLP with assets (x, z) and price bounds $[\gamma, \gamma^{-1}]$ where

$$\gamma = \alpha^{2/3} \cdot \frac{y+a}{L} = \alpha^{2/3} \cdot \left(\frac{y}{L} + \alpha^{1/3}\right). \tag{11}$$

Proof. The marginal trading curve of the 3-CLP is

$$\begin{aligned} & (x_1 + a)(y + a)(z_1 + a) = L^3 \\ \Leftrightarrow \quad & (x_1 + a)(z_1 + a) = \frac{L^3}{y+a} \end{aligned}$$

The trading curve of the above-mentioned 2-CLP is

$$(x_1 + \gamma^{1/2} L')(z_1 + \gamma^{1/2} L') = L'^2$$

where L' is such that it solves the equation for $x_1 = x$ and $z_1 = z$. To show that the two curves are equal, it is sufficient to show that $L' = \sqrt{\frac{L^3}{y+a}}$ is this solution.[19] To see this, note that the offset on the left-hand side is

$$\gamma^{1/2} L' = \sqrt{\gamma \frac{L^3}{y+a}} = \sqrt{\alpha^{2/3} L^2} = \alpha^{1/3} L = a$$

and the right-hand side is

$$L'^2 = \frac{L^3}{y+a} = \frac{(x+a)(y+a)(z+a)}{y+a} = (x+a)(z+a). \qquad \square$$

Intuitively, the above theorem implies that the higher the contribution of asset y to the invariant L compared to the other assets, the tighter the price bounds of the marginal trading curve will be. We cannot easily state this relationship analytically because the closed-form expression for L is rather complicated. However, we can understand some special cases analytically:

Proposition 14. *Consider the price bound parameter γ of the marginal 2-CLP curve from Theorem 2. Then the following hold:*

1. *If $y = 0$, then $\gamma = \alpha$.*
2. *If $x = y = z$, then $\gamma = \alpha^{2/3}$.*
3. *In the limit for $\frac{y}{x}, \frac{y}{z} \to \infty$, $\gamma \to 1$.*

Proof. 1. This follows directly from the second form in (11).

2. If $x = y = z$, then (4) is equivalent to

$$(y + a)^3 = L^3$$

[19] Recall that the solution is always unique.

and thus $\frac{y+a}{L} = 1$. The statement now follows from the first form in (11).

3. First recall that, when all assets are scaled by some factor, then this also scales L by the same factor and the factor cancels out in the term $\frac{y+a}{L}$. Therefore, we can WLOG assume that y is constant and $x, z \to 0$. By continuity, it is enough to consider γ for the limit case where $x = z = 0$.[20] Here, (4) simplifies to

$$\begin{aligned} & & a^2(y+a) &= L^3 \\ &\Leftrightarrow & \alpha^{2/3}L^2(y+a) &= L^3 \\ &\Leftrightarrow & \frac{y+a}{L} &= \alpha^{-2/3}. \end{aligned}$$

The statement now follows from (11). □

2-CLP vs 3-CLP and Capital Efficiency. We can use the above results to compare two alternative ways of providing liquidity. Assume that we want to provide liquidity in two asset pairs: X–Z and Y–Z. Assume that prices in both of these pairs are usually around 1 and we want trades to be on a 2-CLP curve with price bounds $[\gamma, \gamma^{-1}]$ and with a market depth of $x = y = z$ in all assets. We have at least two options to do this:

1. Create two 2-CLP pools with price bounds $[\gamma, \gamma^{-1}]$, one for the X–Z pair and one for the Y–Z pair, and initialize them with equal amounts of each asset, so that the prices in both pools are 1. The total capital required for this is x units of asset X and Y, respectively, and $2x$ units of asset Z.
2. Create a single 3-CLP pool with price bound parameter $\alpha = \gamma^{3/2}$ and initialize it with equal amounts of each asset, so that both the x–z and y–z relative price are 1. By Proposition 14, this will yield the desired price bounds for the x–z and y–z marginal trading curves whenever trading starts at a point where the three assets have equal amounts (i.e., where both relative prices are 1). The total capital required is of course x units of asset X, Y, and Z, respectively, i.e., we save 50% of the capital for asset Z.

The 3-CLP pool requires less capital of asset Z because it uses its asset-Z capital for trading against both asset X and Y. We receive essentially the same trading behavior as with the two 2-CLPs, but with less capital use, if a few assumptions are satisfied:

1. The relative prices in the pool are close to 1 most of the time.
2. Traders do not want to trade large amounts of both assets Y and Z in the same direction at the same time.

[20] Of course, for $x = z = 0$, there is no marginal trading curve and the question regarding price bounds is moot. However, the limiting γ still exists and price bounds of the trading curves for positive x, z will converge to this limit. To receive continuity of the whole operation, we formally need to use the previously-proven fact that (4) has a unique non-negative solution, and we use continuity of both sides of this equation.

These assumptions will be satisfied for many market making use cases, where trades are generally small, arbitrage is quick, and traders and arbitrageurs trade roughly equally in both directions. However, it should be noted that the 3-CLP offers less asset-Z liquidity overall and will therefore be exhausted more quickly when large correlated trades are performed in both asset X and Y.

We can extend the argument to an analogous setting where we want to allow trading between all three asset pairs, X–Y, Y–Z, and X–Z. In this case, by the argument above, the 3-CLP enables 50% savings in capital in each of the three assets X, Y, and Z compared to a setup with three 2-CLP pools.

References

Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. Technical report, Tech. rep., Uniswap, March 2021. URL `https://uniswap.org/whitepaper-v3.pdf`.

Guillermo Angeris and Tarun Chitra. Improved price oracles: Constant function market makers. *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, Oct 2020. URL `http://dx.doi.org/10.1145/3419614.3423251`.

Ariah Klages-Mundt and Steffen Schuldenzucker. Prices in higher-dimensional and transformed constant-function market makers. mimeo, 2021.

Disclaimer