Achieving a maximum and targeted long trade with Hyperdrive

Dylan Paiton

February 2024

Abstract

We first outline the core invariant derivation from price for the modified Yieldspace curve used in Hyperdrive. We then derive the algorithm for computing the maximum long trade amount that is possible for a given Hyperdrive market pool. Finally, we derive a targeted long trade amount for adjusting the Hyperdrive market to a desired fixed rate. The targeted long proposal is also summarized in Hyperdrive #775. The issues we found along the way are summarized in this audit report, Hyperdrive #774.

1 Max long

1.1 The Hyperdrive-Yieldspace AMM

For a deployed market pool, the Hyperdrive-Yieldspace AMM uses a modified constant power sum formula to derive a price relationship between two assets. In this case, our assets are vault shares, z, and bonds, y. When base, x, is supplied to the market, it is converted into shares, z, by depositing the base into an underlying yield bearing vault. The Hyperdrive AMM then supplies bonds, y, such that k is kept constant. The two are related via an invariance formula (yieldspace.rs, l285):

$$\begin{aligned} k &= \frac{\mu}{c}^{-T_s} x^{1-T_s} + y^{1-T_s} \\ &= \frac{c}{\mu}(\mu z)^{1-T_s} + y^{1-T_s} \end{aligned} \tag{1}$$

where T_s is the time stretch constant, c is the current vault share price, and μ is the share price of the vault when the Hyperdrive pool was created (aka `initial_share_price`).

The relationship between shares and bonds is also described using the spot price. Our generic equation for spot price (max.rs, l154 and yieldspace.rs, l36) is given by:

$$p = \left(\frac{\mu z}{y}\right)^{T_s} \tag{2}$$

💡 **NOTE**

We can use the procedure outlined in Appendix C of the YieldSpace paper to relate the price and invariant formula. Recall from the paper:

> In any invariant-based liquidity provision formula, the price at any point along the curve is equal to the negation of the derivative at that point
>
> $p_x = -\frac{dy}{dx}$

However, in Hyperdrive we consider a price that is <= 1, while the YieldSpace paper assumes it is >= 1, which we can express as an inversion: $p_{\text{hyperdrive}} = \frac{1}{p_{\text{yieldspace}}}$. Given this, and that base is converted to shares via $x = cz$, we can derive the invariant from the price as such:

$$
\begin{aligned}
-\tfrac{dy}{dx} &= p^{-1} \\
-\tfrac{dy}{dx} &= \left(\tfrac{\mu z}{y}\right)^{-T_s} \\
-\tfrac{dy}{dx} &= \left(\mu \tfrac{x}{c}\right)^{-T_s} y^{T_s} \\
-y^{-T_s}\tfrac{dy}{dx} &= \left(\mu \tfrac{x}{c}\right)^{-T_s} \\
-\int y^{-T_s} dy &= \int \left(\mu \tfrac{x}{c}\right)^{-T_s} dx \\
-\tfrac{1}{1-T_s} y^{1-T_s} + \alpha_1 &= \mu^{-T_s} \tfrac{1}{c^{-T_s}} \left(\tfrac{1}{1-T_s} x^{1-T_s} + \alpha_2\right) \\
-y^{1-T_s} + \alpha_1' &= \left(\tfrac{\mu}{c}\right)^{-T_s} x^{1-T_s} + \alpha_2'' \\
\alpha_1' - \alpha_2'' &= \left(\tfrac{\mu}{c}\right)^{-T_s} x^{1-T_s} + y^{1-T_s} \\
k &= \tfrac{c}{\mu} \left(\tfrac{\mu}{c}\right)^{1-T_s} x^{1-T_s} + y^{1-T_s} \\
k &= \tfrac{c}{\mu} \left(\tfrac{\mu}{c} x\right)^{1-T_s} + y^{1-T_s} \\
k &= \tfrac{c}{\mu} (\mu z)^{1-T_s} + y^{1-T_s}
\end{aligned}
$$

Hyperdrive computes fees that are removed from the system whenever a trade is made. The fee constants are denoted with ϕ, where ϕ_f refers to the flat fee, ϕ_c refers to the curve fee, and ϕ_g is the governance fee. The open-long governance and curve fees can be written as a function of the base transferred, Δx, and initial spot price (fees.rs):

$$\Phi_c(\Delta x) = \phi_c \left(\frac{1}{p_0} - 1\right) \Delta x \tag{3}$$

$$\Phi_g(\Delta x) = \phi_g p_0 \Phi_c(\Delta x) = \phi_g \phi_c \left(1 - p_0\right) \Delta x \tag{4}$$

where p_0 is the spot price before the trade, i.e. the current spot price. We do not include a function for the flat fee because it is only applied when closing a long.

> 💡 **NOTE**
> We will use capital letters to denote functions and lower-case letters to denote scalars.

The pool's maximum spot price such that the trade doesn't result in negative interest is given by (max.rs, line 147 and derived in issue #655):

$$p_{\max} = \frac{(1-\phi_f)}{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)} \tag{5}$$

1.2 Deriving the target base and bond amounts

Our goal is to determine the maximum long that can be opened for a given market, which will result in the max spot price. The two price equations can be used to derive the target reserve levels for a pool with the max spot price. These are given as target shares, z_t, and target bonds, y_t. First we will solve for the target bond reserves in terms of the target share reserves by setting equations (2) and (5) to be equal max.rs, l162:

$$\begin{aligned}
p &= p_{\max} \\
\left(\frac{\mu z_t}{y_t}\right)^{T_s} &= \frac{(1-\phi_f)}{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)} \\
\frac{\mu z_t}{y_t} &= \left(\frac{(1-\phi_f)}{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}\right)^{\frac{1}{T_s}} \\
y_t &= \frac{\mu z_t}{\left(\frac{(1-\phi_f)}{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}\right)^{\frac{1}{T_s}}} \\
y_t &= \mu z_t \left(\frac{1+\phi_c(\frac{1}{p_0}-1)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1}{T_s}}
\end{aligned} \tag{6}$$

> 💡 **NOTE** That last step required some algebra acrobatics:
>
> $$\frac{1}{\left(\frac{x}{y}\right)^{c}} = \frac{1}{\frac{x^c}{y^c}} = \frac{1}{\frac{y^{-c}}{x^{-c}}} = \frac{1}{\left(\frac{y}{x}\right)^{-c}} = \left(\frac{y}{x}\right)^{c}$$

Using the invariant equation we can solve for z_t isolated, without y_t max.rs, l175:

$$
\begin{aligned}
k &= \tfrac{c}{\mu}(\mu z_t)^{1-T_s} + y_t^{1-T_s} \\
k &= \tfrac{c}{\mu}(\mu z_t)^{1-T_s} + \left(\mu z_t \left(\frac{1+\phi_c(\frac{1}{p_0}-1)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1}{T_s}}\right)^{1-T_s} \\
k &= (\mu z_t)^{1-T_s} \left(\frac{c}{\mu} + \left(\frac{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1-T_s}{T_s}}\right) \\
(\mu z_t)^{1-T_s} &= k \Bigg/ \left(\frac{c}{\mu} + \left(\frac{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1-T_s}{T_s}}\right) \qquad (7) \\
\mu z_t &= \left(k \Bigg/ \left(\frac{c}{\mu} + \left(\frac{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1-T_s}{T_s}}\right)\right)^{\frac{1}{1-T_s}} \\
z_t &= \frac{1}{\mu}\left(k \Bigg/ \left(\frac{c}{\mu} + \left(\frac{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1-T_s}{T_s}}\right)\right)^{\frac{1}{1-T_s}}
\end{aligned}
$$

Next, we plug this result into our earlier equation to get y_t isolated (max.rs, l202):

$$
\begin{aligned}
y_t &= \mu z_t \left(\frac{1+\phi_c(\frac{1}{p_0}-1)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1}{T_s}} \\
&= \left(k \Bigg/ \left(\frac{c}{\mu} + \left(\frac{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1-T_s}{T_s}}\right)\right)^{\frac{1}{1-T_s}} \left(\frac{1+\phi_c\left(\frac{1}{p_0}-1\right)(1-\phi_f)}{1-\phi_f}\right)^{\frac{1}{T_s}}
\end{aligned} \tag{8}
$$

These target reserve levels then correspond to opening a long for a delta base or bonds (max.rs, l213 and max.rs, l219, respectively):

$$
\Delta x = c(z_t - z) \tag{9}
$$

$$
\Delta y = (y - y_t) - \Phi_c(\Delta x) \tag{10}
$$

If the pool is solvent after opening this long, then we're done. Otherwise, we will use a numerical approach to estimate the actual trade amount.

1.3 Iterative refinement of the maximum long amount

Opening a long causes a change in both base and bonds and is impacted by fees. Without a closed-form solution, we will need a numerical approach to estimate the actual trade amount for most pool conditions. Specifically, we will use Newton's method with the pool's solvency as our objective function. Solvency captures the protocol's ability to pay its debts by measuring its assets versus its liabilities and minimum reserves. Assets are the share reserves, z. Liabilities are the aggregate long exposure. Minimum share reserves are set in a pool's configuration, as z_{min}. Liabilities and reserves are converted to common units (base or shares) via the share price, c.

$$\begin{aligned} S(z) &= \text{assets} - \text{liabilities} - \text{minimum\_reserves} \\ &= z - \tfrac{l}{c} - z_{\text{min}} \\ &= \tfrac{1}{c}\left(x - l - x_{\text{min}}\right) \end{aligned} \tag{11}$$

For a single long, the change in exposure is given by the amount of bonds returned, $\Delta l = Y(\Delta x)$ (aka the amount of longs opened, or long amount). The amount of bonds returned can be broken down into a component without fees and a fee component (open.rs, l11):

$$Y(\Delta x) = Y_*(\Delta x) - \Phi_c(\Delta x) \tag{12}$$

where, for some initial bond reserves, y_0, and base reserves, x_0 (or alternatively initial effective share reserves, z_0),

$$\begin{aligned} Y_*(\Delta x) &= y_0 - \left(k - \tfrac{c}{\mu}\left(\mu\left(z_0 + \tfrac{\Delta x}{c}\right)\right)^{1-T_s}\right)^{\frac{1}{1-T_s}} \\ &= y_0 - \left(k - \tfrac{c}{\mu}\left(\tfrac{\mu}{c}\left(x_0 + \Delta x\right)\right)^{1-T_s}\right)^{\frac{1}{1-T_s}} \\ &= y_0 - \left(k - \left(\tfrac{\mu}{c}\right)^{-T_s}\left(x_0 + \Delta x\right)^{1-T_s}\right)^{\frac{1}{1-T_s}} \end{aligned} \tag{13}$$

When a long is opened, the share reserves is increased by (max.rs, l315):

$$\Delta z = \tfrac{\Delta x - \Phi_g(\Delta x)}{c} \tag{14}$$

Using these components, we can derive our objective function as the solvency after a change in shares (max.rs, l329):

$$\begin{aligned} S(\Delta z) &= (z_0 + \Delta z) - \left(\tfrac{l_0 + l_{\text{chk}} + \Delta l}{c}\right) - z_{\text{min}} \\ &\therefore \\ S(\Delta x) &= \tfrac{1}{c}\left(x_0 + \Delta x - \Phi_g\left(\Delta x\right) - l_0 - l_{\text{chk}} - Y(\Delta x) - x_{\text{min}}\right) \end{aligned} \tag{15}$$

where l_{chk} is the checkpoint long exposure and is assumed to be $>= 0$. We add the checkpoint exposure to account for negative exposure from non-netted shorts in the checkpoint. We will keep everything in units of base, but see max.rs, l336 for the implementation using units of shares.

In summary, our optimization objective is

$$\begin{aligned} &\underset{\Delta x}{\operatorname{argmax}} -S(\Delta x) \\ &\text{s.t.} S(\Delta x) > 0 \end{aligned} \tag{16}$$

To perform Newton's method we also need the gradient of the objective, starting with the long amount:

$$Y'(\Delta x) = Y'_*(\Delta x) - \Phi'_c(\Delta x) \tag{17}$$

where,

$$\begin{aligned} Y'_*(\Delta x) &= \left(\tfrac{\mu}{c}\right)^{-T_s} (x_0 + \Delta x)^{-T_s} \left(k - \left(\tfrac{\mu}{c}\right)^{-T_s} (x_0 + \Delta x)^{1-T_s}\right)^{\frac{T_s}{1-T_s}} \\ &= \big(\mu(z_0 + \tfrac{\Delta x}{c})\big)^{-T_s} \left(k - \tfrac{c}{\mu} \big(\mu(z_0 + \tfrac{\Delta x}{c}\big)^{1-T_s}\right)^{\frac{T_s}{1-T_s}} \end{aligned} \tag{18}$$

We also need the gradient of the governance and curve fee calculations:

$$\Phi'_c(\Delta x) = \phi_c(\tfrac{1}{p_0} - 1) \tag{19}$$

$$\Phi'_g(\Delta x) = \phi_g p_0 \Phi'_c(\Delta x) \tag{20}$$

Together, these give us the solvency gradient:

$$S'(\Delta x) = \tfrac{1}{c} \big(1 - \Phi'_g(\Delta x) - Y'(\Delta x)\big) \tag{21}$$

We want to discover a Δx to push the pool to be as close to insolvent as possible, without passing over to actually being insolvent. We achieve this by maximizing the negative solvency, since solvency decreases as more longs are opened. For each iteration of Newton's method (max.rs, l73):

$$\begin{aligned} \Delta x_{n+1} &= \Delta x_n - \tfrac{S(\Delta x_n)}{S'(\Delta x_n)} \\ &= \Delta x_n + \tfrac{S(\Delta x_n)}{-S'(\Delta x_n)} \end{aligned} \tag{22}$$

In the actual implementation, we will iteratively compute solvency for the new Δx_n until the system is no longer solvent, and then back up one step to return the maximum long.

1.4 Deriving an initial guess for the max long amount

The rate of convergence for Newton's method is improved with a better initial guess, $\Delta x_{n=0}$. To derive an initial guess, we can use a conservative price estimate, p_r, to approximate $Y(\Delta x)$ (max.rs, l253):

$$\begin{aligned} Y(\Delta x) &\approx \tfrac{\Delta x}{p_r} - \Phi_c(\Delta x) \\ &\approx \Delta x \left(\tfrac{1}{p_r} - \phi_c \left(\tfrac{1}{p_0} - 1 \right) \right) \end{aligned} \tag{23}$$

We define our initial solvency as in equation (11): $s_0 = \frac{1}{c}(x_0 - l_0 - x_{\text{min}})$. Plugging this into our solvency function $S(\Delta x)$, we can calculate the share reserves and exposure after opening a long with Δx base as (max.rs, l259):

$$\begin{aligned} Z(\Delta x) &= z_0 + \tfrac{\Delta x - g(\Delta x)}{c} \\ Z(\Delta x) &= \tfrac{1}{c}(x_0 + \Delta x - \Phi_g(\Delta x)) \end{aligned} \tag{24}$$

$$\begin{aligned} L(\Delta x) &= l_0 + l_{\text{chk}} + 2Y(\Delta x) - \Delta x + \Phi_g(\Delta x) \\ &= l_0 + l_{\text{chk}} + 2p_r^{-1}\Delta x - 2\Phi_c(\Delta x) - \Delta x + \Phi_g(\Delta x) \end{aligned} \tag{25}$$

These formulae allow us to calculate the approximate ending solvency of (max.rs, l271):

$$\begin{aligned} S(\Delta x) &\approx Z(\Delta x) - \tfrac{L(\Delta x)}{c} - z_{\text{min}} \\ &\approx \tfrac{1}{c}(x_0 + \Delta x - \Phi_g(\Delta x) - L(\Delta x) - x_{\text{min}}) \\ &\approx \tfrac{1}{c}\left(x_0 + 2\Delta x - 2\Phi_g(\Delta x) - l_0 - l_{\text{chk}} - 2\tilde{Y}(\Delta x) - x_{\text{min}} \right) \end{aligned} \tag{26}$$

If we rearrange to represent the initial solvency, s_0, then we can solve for Δx (max.rs, l278):

$$\begin{aligned} s_0 - \tfrac{1}{c}l_{\text{chk}} + \tfrac{2}{c}\left(\Delta x - \Phi_g(\Delta x) - \tilde{Y}(\Delta x) \right) &\approx 0 \\ s_0 - \tfrac{1}{c}l_{\text{chk}} + \tfrac{2}{c}\left(\Delta x - \Phi_g(\Delta x) - \tilde{Y}(\Delta x) \right) &\approx 0 \\ s_0 - \tfrac{1}{c}l_{\text{chk}} + \tfrac{2}{c}\Delta x \left(1 - \phi_g\phi_c(1 - p_0) - \tfrac{1}{p_r} + \phi_c\left(\tfrac{1}{p_0} - 1 \right) \right) &\approx 0 \end{aligned} \tag{27}$$

$$\therefore$$

$$\begin{aligned} \Delta x &\approx \tfrac{c}{2} \frac{-(s_0 - (\frac{1}{c}l_{\text{chk}}))}{1 - \phi_g\phi_c(1-p_0) - p_r^{-1} + \phi_c\left(p_0^{-1} - 1\right)} \\ \Delta x &\approx \tfrac{c}{2} \frac{s_0 + \frac{1}{c}l_{\text{chk}}}{p_r^{-1} + \phi_g\phi_c(1-p_0) - \phi_c\left(p_0^{-1} - 1\right) - 1} \end{aligned} \tag{28}$$

This gives us the initial value for Δx_0 in the iterative process.

> ❗ **Discrepancy**
>
> This does not match max.rs, l73, which has the numerator equal to $s_0 + l_{\text{chk}}$.

2 Targeted long

2.1 Targeted long for a given rate

We can follow a similar derivation to get a long that results in a *target* fixed rate (aka spot rate). The fixed rate for a Hyperdrive pool given the spot price, p, and the annualized position duration, $\bar{d}$ is given by:

$$r = (1 - p)(p\bar{d})^{-1} \tag{29}$$

Note that the conversion from a "price" (which is computed at a single point in time) to a "rate" (which is computed using at least 2 points in time) is automatic because of the predetermined position duration. Solving for p, we get:

$$\begin{aligned}
(1 - p)(p\bar{d})^{-1} &= r \\
\tfrac{1}{p\bar{d}} - \tfrac{1}{\bar{d}} &= r \\
\tfrac{1}{p}\tfrac{1}{\bar{d}} &= r + \tfrac{1}{\bar{d}} \\
\tfrac{1}{p} &= d\left(r + \tfrac{1}{\bar{d}}\right) \\
p &= \left(r\bar{d} + 1\right)^{-1}
\end{aligned} \tag{30}$$

As before, we use $p = \left(\frac{\mu z_t}{y_t}\right)^{T_s}$ to find target reserve bonds, y_t, in terms of a target rate, r_t (utils.rs, l112):

$$\begin{aligned}
\left(\tfrac{\mu z_t}{y_t}\right)^{T_s} &= \tfrac{1}{r_t\bar{d}+1} \\
\tfrac{\mu z_t}{y_t} &= \left(\tfrac{1}{r_t\bar{d}+1}\right)^{\frac{1}{T_s}} \\
y_t &= \frac{\mu z_t}{\left(\frac{1}{r_t\bar{d}+1}\right)^{\frac{1}{T_s}}} \\
y_t &= \mu z_t\left(r_t\bar{d} + 1\right)^{\frac{1}{T_s}}
\end{aligned} \tag{31}$$

We then use the invariant formula from Equation (1) to determine the share reserves required for a given rate.

$$\begin{aligned}
\frac{c}{\mu}(\mu z_t)^{1-T_s} + y_t^{1-T_s} &= k \\
\frac{c}{\mu}(\mu z_t)^{1-T_s} + \left(\mu z_t \left(r_t \bar{d} + 1\right)^{\frac{1}{T_s}}\right)^{1-T_s} &= k \\
\frac{c}{\mu}(\mu z_t)^{1-T_s} + (\mu z_t)^{1-T_s} \left(\left(r_t \bar{d} + 1\right)^{\frac{1}{T_s}}\right)^{1-T_s} &= k \\
(\mu z_t)^{1-T_s} \left(\frac{c}{\mu} + \left(\left(r_t \bar{d} + 1\right)^{\frac{1}{T_s}}\right)^{1-T_s}\right) &= k \\
(\mu z_t)^{1-T_s} &= \frac{k}{\frac{c}{\mu} + \left(\left(r_t \bar{d} + 1\right)^{\frac{1}{T_s}}\right)^{1-T_s}} \\
z_t &= \frac{1}{\mu} \left(\frac{k}{\frac{c}{\mu} + \left(\left(r_t \bar{d} + 1\right)^{\frac{1}{T_s}}\right)^{1-T_s}}\right)^{\frac{1}{1-T_s}}
\end{aligned} \tag{32}$$

And finally, we plug this in to equation (31) to isolate the target bonds, y_t.

$$y_t = \left(\frac{k}{\frac{c}{\mu} + \left(\left(r_t \bar{d} + 1\right)^{\frac{1}{T_s}}\right)^{1-T_s}}\right)^{1-T_s} \left(r_t \bar{d} + 1\right)^{\frac{1}{T_s}} \tag{33}$$

Using these targets in Equations (9) and (10), we can compute the long base amount to hit a target rate assuming an infinitesimally-derived price (i.e. spot price). The approximate reserve levels for a target rate are much more likely to be solvent than the reserves after an approximated maximum long, but we still have to deal with the discrepancy between the spot price and the realized price that arises from a realistic trade size.

2.2 Iteratively finding a trade for a target rate

We need to know how the rate changes when base reserves change, $R(\Delta x)$, which will become our new objective function. The derivative of this objective will give us the updates in each step of the refinement algorithm.

Picking up from (12), we can write the full equation for the bonds received for a given base provided from a long trade:

$$Y(\Delta x) = y_0 - \left(k - \left(\frac{\mu}{c}\right)^{-T_s} (x_0 + \Delta x)^{1-T_s}\right)^{\frac{1}{1-T_s}} - \phi_c \left(\frac{1}{p_0} - 1\right) \Delta x \tag{34}$$

This is also the amount that is subtracted from the pool, i.e. $\Delta y_{\text{pool}} = Y(\Delta x)$ and thus $y_{\text{new-pool}} = y_{\text{old-pool}} - Y(\Delta x)$. The corresponding delta that would be applied to the pool's effective share reserves is:

$$\begin{aligned} \Delta z_{e,\text{pool}} &= \tfrac{\Delta x}{c} - \Phi_g(\Delta x) - \zeta \\ &= \tfrac{1}{c}\left(\Delta x - \phi_g \phi_c \left(1 - p_0\right)\right) - \zeta \end{aligned} \tag{35}$$

where ζ is the pool's zeta adjustment state, which is unchanged when opening positions. The instantaneous spot price given pool reserve levels (z_e, y) is

$$p = \left(\tfrac{\mu z_e}{y}\right)^{T_s} \tag{36}$$

Together these allow us to derive the new share price after opening a long (open.rs, l52):

$$\begin{aligned} P(\Delta x) &= \left(\tfrac{\mu(z_{e0,\text{pool}} + \Delta z_{e,\text{pool}})}{(y - \Delta y_{\text{pool}})}\right)^{T_s} \\ P(\Delta x) &= \left(\frac{\mu\left(z_{e0,\text{pool}} + \frac{\Delta x}{c} - \Phi_g(\Delta x) - \zeta\right)}{(y - Y(\Delta x))}\right)^{T_s} \\ &= \left(\frac{\mu\left(z_{e0} + \frac{1}{c}(\Delta x - \phi_g \phi_c (1 - p_0)) - \zeta\right)}{\left(y_0 - \left(y_0 - \left(k - \left(\frac{\mu}{c}\right)^{-T_s} (x_0 + \Delta x)^{1 - T_s}\right)^{\frac{1}{1 - T_s}} - \phi_c \left(\frac{1}{p_0} - 1\right) \Delta x\right)\right)}\right)^{T_s} \end{aligned} \tag{37}$$

where again p_0, y_0, z_{e0}, and x_0 are the spot price, bond reserves, effective share reserves, and base reserves before the trade, respectively.
We will also need the derivative of this function:

$$\begin{aligned} P'(\Delta x) = T_s &\left(\frac{y_0 - Y(\Delta x)}{\mu\left(z_{e0} + \frac{\Delta x}{c} - \Phi_g(\Delta x) - \zeta\right)}\right)^{1 - T_s} \\ &\frac{(y_0 - Y(\Delta x))\left(\frac{\mu}{c} - \Phi'_g(\Delta x)\right) + Y'(\Delta x)\mu\left(z_{e0} + \frac{\Delta x}{c} - \Phi_g(\Delta x) - \zeta\right)}{(y_0 - Y(\Delta x))^2} \end{aligned} \tag{38}$$

Given this, we can write the rate:

$$R(\Delta x) = (1 - P(\Delta x))\left(P(\Delta x)\bar{d}\right)^{-1} \tag{39}$$

And the derivative of the rate:

$$\begin{aligned} R'(\Delta x) &= \frac{-P'(\Delta x)P(\Delta x)\bar{d} - (1 - P(\Delta x))\left(P'(\Delta x)\bar{d}\right)}{(P(\Delta x)\bar{d})^2} \\ &= \frac{-P'(\Delta x)P(\Delta x)\bar{d} - P'(\Delta x)\bar{d} + P'(\Delta x)P(\Delta x)\bar{d}}{(P(\Delta x)\bar{d})^2} \\ &= \frac{-P'(\Delta x)}{P(\Delta x)^2\bar{d}} \end{aligned} \tag{40}$$

We can now write our optimization function for the Newton updates. $l(\Delta x) = R(\Delta x) - r_t$ shifts the trading curve down towards the zero-point.

$$\begin{aligned} &\underset{\Delta x}{\text{argmax}}(R(\Delta x) - r_t) \\ &\quad \text{s.t.} S(\Delta x) > 0 \end{aligned} \tag{41}$$

As before, the derivative of loss gives us our Δx:

$$\begin{aligned} \Delta x_{n+1} &= \Delta x - \frac{l(\Delta x)}{l'(\Delta x)} \\ &= \Delta x_n - \frac{R(\Delta x) - r_t}{R'(\Delta x)} \\ &= \Delta x_n + \frac{R(\Delta x) - r_t}{-R'(\Delta x)} \end{aligned} \tag{42}$$