Platypus - Potential for Arbitrage

George Weatherill, Haruko Engineer

Overview

In this article we present a detailed analysis of the mathematical underpinnings of the Platypus cryptocurrency exchange platform (Platypus Team, 2022), specifically its arbitrage protection. During its development the Platypus development team identified the potential for arbitrage and explicitly include penalties for liquidity withdrawals or deposits, over and above swap price slippages, trying to exclude such attack vectors on Platypus's liquidity.

Through a full, step by step algebraic analysis of potential arbitrage routes several significant issues in the arbitrage protection methodology outlined in the Platypus Yellow Paper (Platypus Team, 2022, p. 19 & 22) are identified: neglecting "slippage of slippage", failure to properly reverse swaps, applying deposit or withdrawal penalties in specific instances. Together these combine to completely undermine many of the arbitrage related results within the paper.

Based on this analysis, we demonstrate that arbitrage *is* possible on Platypus and that the only obstruction to arbitrage in a StableSwap implementation of Platypus is the 0.01% "haircut fees", not the deposit and withdrawal penalties specifically designed to prevent arbitrage.

Through numerical simulation we demonstrate that if the platform lowered its haircut fees below 0.0067%, a reduction of just 0.33 basis points, arbitrage would be actively possible.

The document breaks down as follows:

- Platypus - Brief recap of the exchange's model, notation, definitions
- Proof of Arbitrage – Detailed mathematical proof of viable attacks on Platypus
- Exploration of Arbitrage – Numerical and qualitative assessment of attacks
- Summary – Brief overview of results and potential changes Platypus could implement

Though some aspects will be recapped or elaborated on, this article assumes the reader is comfortable with mathematical notation and has a good working familiarity with the Platypus AMM Technical Specifications Yellow Paper (Platypus Team, 2022) document, particularly the slippage formalism. Though dense in mathematical notation, this article uses no mathematical concepts beyond those of the Yellow Paper, largely restricting itself to algebraic manipulation and basic calculus.

For the non-mathematically reader, or one just short on time, the *Summary* section contains all primary results and conclusions.

This article is an offshoot of an article exploring multiple DeFi exchanges (Haruko, 2022), including UniSwap, Curve, Bancor, Balancer and Platypus itself. This article serves as deep-dive into Platypus necessary to demonstrate Platypus-related conclusions stated in the original article.

Platypus Model

Though much of the notation used is identical to the Platypus Yellow Paper there are several important additions or modifications for generality or to aid in highlighting issues. A reader already very familiar with the Platypus notation should still briefly read this section.

Parameters

Platypus tracks the status of each token types T_i using 4 parameters:

- A_i – The currently held liquidity for T_i available for swaps or withdrawals
- L_i – The liability the pool has for T_i, the net total of all T_i deposits minus all T_i withdrawals
- r_i – The coverage ratio $\frac{A_i}{L_i}$. Values below 1 are "under-covered", above 1 are "over-covered"
- p_i – The price of a single T_i token relative to some numeraire, obtained from an oracle
- h – Swap fees

Actions

A swap where a trader sells Δ_i units of T_i to the pool in exchange for Δ_j units[1] of T_j affects assets and coverage but not liabilities:

$$\text{Swap} \quad : \quad A_i \to A_i + \Delta_i \quad , \quad A_j \to A_j - \Delta_j \quad , \quad r_i \to \frac{A_i + \Delta_i}{L_i} \quad , \quad r_j \to \frac{A_j - \Delta_j}{L_j}$$

Calculating the relationship between swap quantities $\Delta_i,\ \Delta_j$ requires taking account of price slippage and fees. Swaps between two such tokens do not alter the pool's associated liabilities; this only occurs during deposits or withdrawals.

Swaps are *bilateral*, impacting two tokens, in contrast to deposits and withdrawals, that are *unilateral* as they impact only a single token.

[1] Note that in this representation $\Delta_{i,j} > 0$. This is the "human perspective". From the "pool perspective" $dA_j = -\Delta_j < 0$ as it decreases its T_j holdings.

A deposit of D_i units of T_i affects all factors and can incur a penalty α_i.

$$\text{Deposit} \quad : \quad A_i \to A_i + D_i \quad , \quad L_i \to L_i + (D_i - \alpha_i) \quad , \quad r_i \to \frac{A_i + D_i}{L_i + (D_i - \alpha_i)}$$

When a deposit is made the pool mints its own corresponding tokens to facilitate the liquidity provider performing a withdrawal later. In effect[2] a swap is performed where the "out token" is specific to the pool's smart contract. A pool minted token corresponding to 1 T_1 we denote as C_1. The value of the minted tokens is equal to the increase in liability after the penalty has been accounted for. A deposit of D_i units of T_i is therefore expressible as the swap $D_i T_i \to (D_i - \alpha_i) C_i$ but we will keep the unilateral/bilateral distinction for convenience.

The penalty α_i depends on the pool status and reduces the number of minted tokens the liquidity provider receives[3], reducing the liability of the pool. We refrain from making this dependency explicit here as it is explored in more depth later.

A withdrawal occurs when such minted tokens are "redeemed" with the pool, where they are burnt. Redeeming W_i units of T_i affects all three factors and can incur a penalty β_i.

$$\text{Withdrawal} \quad : \quad A_i \to A_i - (W_i - \beta_i) \quad , \quad L_i \to L_i - W_i \quad , \quad r_i \to \frac{A_i - (W_i - \beta_i)}{L_i - W_i}$$

The penalty β_i depends on the pool status and reduces the number of tokens the liquidity provider receives[4], effectively increasing the assets of the pool – its specific form is also given shortly.

Penalties and Fees

Due to a combination of slippage-based fees and arbitrage-motivated penalties for certain deposits and withdrawals or the deposit (withdrawals) relative to minted (burnt) tokens is somewhat elaborate.

Swap Slippage

For a swap $\Delta_i \to \Delta_j$ Platypus calculates the Δ_j from Δ_i using the account slippage function $g(r)$ (Platypus Team, 2022, p. 6) and the marginal slippage function $S(r, r')$:

$$S(r, r') = \frac{g(r') - g(r)}{r' - r}$$

[2] The specific mechanism by which Platypus tracks liquidity holdings is not relevant so long as a liquidity provider has a redeemable "IOU" for the appropriate amount after deposit penalties.
[3] A LP depositing 10 T_1 but might receive only enough tokens to redeem 9.995T_1 later.
[4] An LP redeeming smart tokens equivalent to 10 T_1 but might only receive 9.995T_1 from the pool.

A cost-free system would exchange $\Delta_i T_i$ into $\frac{p_i}{p_j}\Delta_i T_j$, where $\frac{p_i}{p_j} \equiv f_{i\to j}$ is the relative price or "exchange rate". The new cost-free coverages would be

$$r_i \to r_i' = \frac{A_i + \Delta_i}{L_i} = r_i + \frac{\Delta_i}{L_i} \quad , \quad r_j \to r_j' = \frac{A_j - f_{i\to j}\Delta_i}{L_j} = r_j - \frac{f_{i\to j}\Delta_i}{L_j}$$

Token-wise marginal slippages are then calculated from the cost-free coverages:

$$S_i = S(r_i, r_i') = \frac{g(r_i') - g(r_i)}{r_i' - r_i} \quad , \quad S_j = S(r_j, r_j') = \frac{g(r_j') - g(r_j)}{r_j' - r_j}$$

The total slippage then follows:

$$S_{i\to j} = S_i - S_j = \left(\frac{g(r_i') - g(r_i)}{r_i' - r_i}\right) - \left(\frac{g(r_j') - g(r_j)}{r_j' - r_j}\right)$$

The swap is then performed according to an updated "terminal exchange rate" $f_{i\to j}^*$ that depends on the slippage and fees:

$$\Delta_j = f_{i\to j}(1 - S_{i\to j})(1 - h)\Delta_i \equiv f_{i\to j}^*\Delta_i$$

$S_{i\to j} > 0$ decreases $f_{i\to j}^*$, penalising the swap, while $S_{i\to j} < 0$ increases $f_{i\to j}^*$, rewarding the trader who performed the swap (fees not withstanding).

The procedure to calculate the "amount out" in a swap is

- Calculate cost-less coverage changes $r_i \to r_i'$, $r_j \to r_j'$
- Calculate $S_i = S(r_i, r_i')$, $S_j = S(r_j, r_j')$ on cost-less coverages
- Calculate terminal exchange rate $f_{i\to j}^*$ and thus Δ_j.

The *true* post-swap coverages are then calculated using the terminal exchange rate:

$$r_i \to r_i' = \frac{A_i + \Delta_i}{L_i} = r_i + \frac{\Delta_i}{L_i} \quad , \quad r_j \to r_j' = \frac{A_j - f_{i\to j}^*\Delta_i}{L_j} = r_j - \frac{f_{i\to j}^*\Delta_i}{L_j}$$

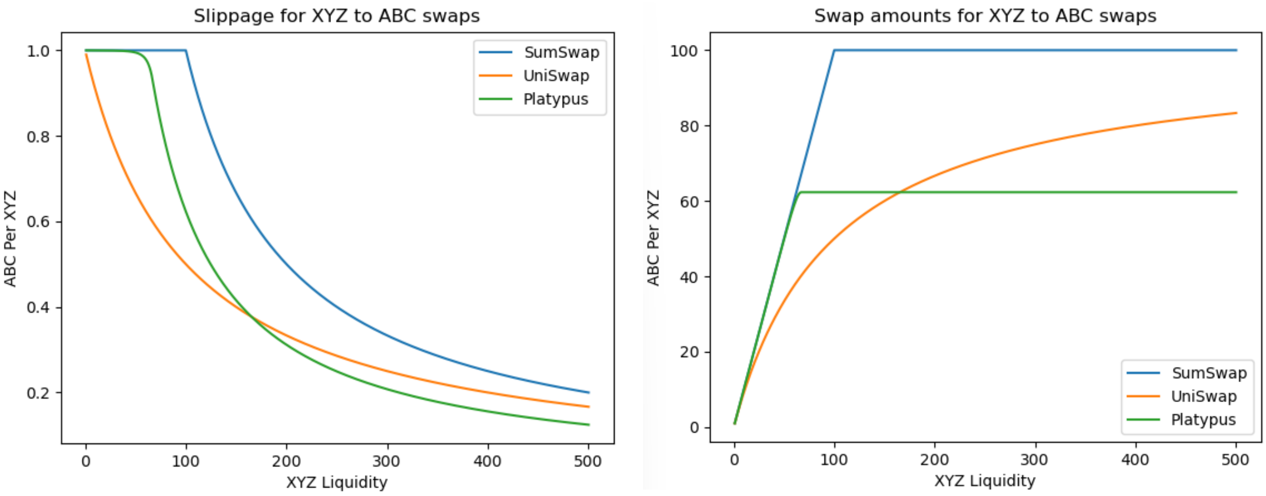

Figure 1 : Slippage impact on prices and volumes for a pool containing 100 ABC and 100 XYZ tokens with 1:1 pricing for UniSwap (Orange), fixed price SumSwap (Blue) and Platypus (Green).

An example of Platypus slippage is given in Figure 1, in comparison to UniSwap and the infinite leverage, fixed pricing SumSwap. Platypus experiences practically zero slippage until the coverage drops to $r \approx r^*$, so the maximum swap volume is $L(1 - r^*)$. At $r = r^*$ the marginal slippage becomes 100%, meaning the liquidity in each token cannot be drained below $r \leq r^*$ through swaps alone, it must be done via withdrawals.

Arbitrage Potential

Platypus acknowledges this slippage model has the potential for arbitrage attacks on its. To address this, the Yellow Paper considers two 3-step procedures – swap then "unilateral action" (withdrawal or deposit) then reverse swap. The penalties for withdrawals and swaps are then set to make sure the value change in the pool is never negative.

Withdrawal Penalty

We will simply quote the Platypus expressions (Platypus Team, 2022, p. 22), since their deeper exploration is the main purpose of this article.

Withdrawing W tokens at coverage r and liabilities L incurs the following penalty

$$\alpha(W|r, L) = \mathrm{Ind}(r < 1)[g(r')(L - W) - g(r)L + g(1)W] \quad , \quad r' = \frac{rL - W}{L - W}$$

Withdrawing moves the coverage *away* from 1. For $r > 1$ this increases the coverage but for $r < 1$ this decreases coverage. As shown above, this is the way to achieve dangerously low coverage, $r \ll r^*$. Since withdrawals on $r > 1$ increase coverage Platypus applies the penalty if, and only if, $r < 1$.

Deposit Penalty

Quoting the Platypus expressions (Platypus Team, 2022, p. 25), depositing D tokens at coverage r and liabilities L incurs the following penalty

$$\beta(D|r,L) = \mathrm{Ind}(r{>}1)[g(r')(L+D) - g(r)L] \quad , \quad r' = \frac{rL+D}{L+D}$$

Depositing moves the coverage *towards* 1. For $r < 1$ this increases the coverage but for $r > 1$ this decreases coverage and Platypus applies the same principles as the withdrawal penalty, applying a deposit penalty if, and only if, $r > 1$.

Uni-Dependent Limited Support Penalty

It is important to note that the penalties for both unilateral actions depend only on the liquidity of the token that is deposited or withdrawn, the liquidity of the other side of the swap is not included.

This, along with the fact the penalties are applied only for specific coverage ranges, is a *critical* error we will explore further by elaborating on the details of a *4-step* procedure.

Arbitrage

To assess the Platypus model of arbitrage we will first construct a model of Platypus swaps and value changes from first principles. Though lengthy, it is necessary to fully evaluate the approach of the Yellow Paper and identify viable attack vectors on Platypus's liquidity.

General Procedure

A general arbitrage procedure looks to perform a sequence of steps that extract value from the pool in such a way it can be repeated multiple times until the pool is no longer capable of outputting further value.

Through Platypus can work with any number of token types via its unilateral liquidity model we focus on swaps, deposits and withdrawals involving a specific pair of tokens, T_i and T_j, as this will be sufficient to demonstrate our results.

To be repeatable a sequence of steps that "closes the loop" are required – each swap should have a counter-swap, each unilateral action a reverse. For 2 tokens there are 4 inequivalent[5] 4-step potential routes to arbitrage[6]:

- Type 1: swap $T_i \rightarrow T_j$, deposit T_i, swap $T_j \rightarrow T_i$, withdraw T_i
- Type 2: swap $T_i \rightarrow T_j$, deposit T_j, swap $T_j \rightarrow T_i$, withdraw T_j
- Type 3: swap $T_i \rightarrow T_j$, withdraw T_i, swap $T_j \rightarrow T_i$, deposit T_i
- Type 4: swap $T_i \rightarrow T_j$, withdraw T_j, swap $T_j \rightarrow T_i$, deposit T_j

In their analysis of deposit and withdrawal arbitrage penalties the Platypus Yellow Paper does 3 steps, skipping over the "reverse unilateral action" 4th step but this does not impact their findings as the 4th steps they consider would not incur a penalty. For completeness we will include the 4th step in our analysis.

To provide some heuristic understanding of how these potential arbitrage routes impact coverage ratios Figure 5 - Figure 5 show multiple (r_1, r_2) evolution scenarios for each type assuming zero fees, penalties, and slippage. Fees, penalties, and slippage will impact the exact coverage values but neglecting these factors is sufficient to grasp generic behaviour during swaps and unilateral actions.

Several important behaviours need to be highlighted:

1. Deposits push coverage *towards* $r = 1$
2. Withdrawals push coverage *away* from $r = 1$
3. Only swaps can cross the $r = 1$ "barrier"

[5] 4 further, but equivalent, route types are obtained by exchanging $T_i \leftrightarrow T_j$ indices in all expressions
[6] More specifically, the potential for all but one quantity to have closed loops of value.

Type 1 and Type 2 are more natural procedures for someone looking to perform arbitrage – they begin from a zero position, while Type 3 and Type 4 require a pre-existing position within the pool to withdraw or to purchase someone else's via a pool's smart token.

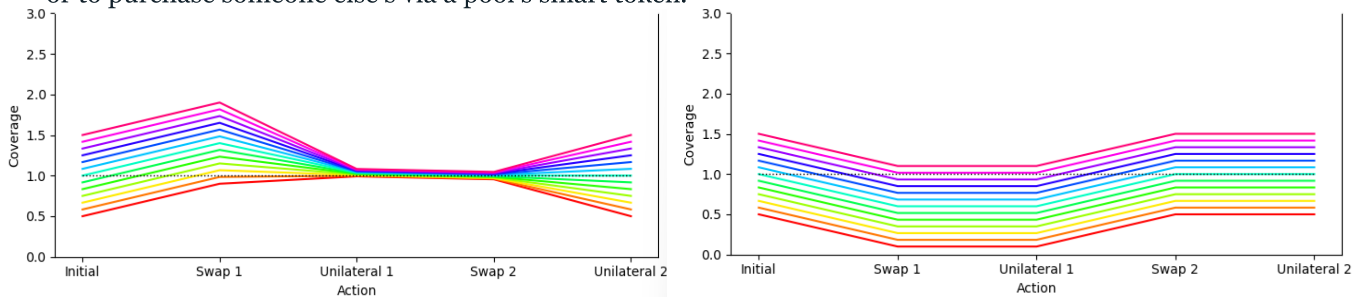

Figure 5 : Coverage evolution for Type 1 routes across multiple generic exchange configurations
Type 1: Swap $T_1 \to T_2$, deposit T_1, reverse swap $T_2 \to T_1$, withdraw T_1

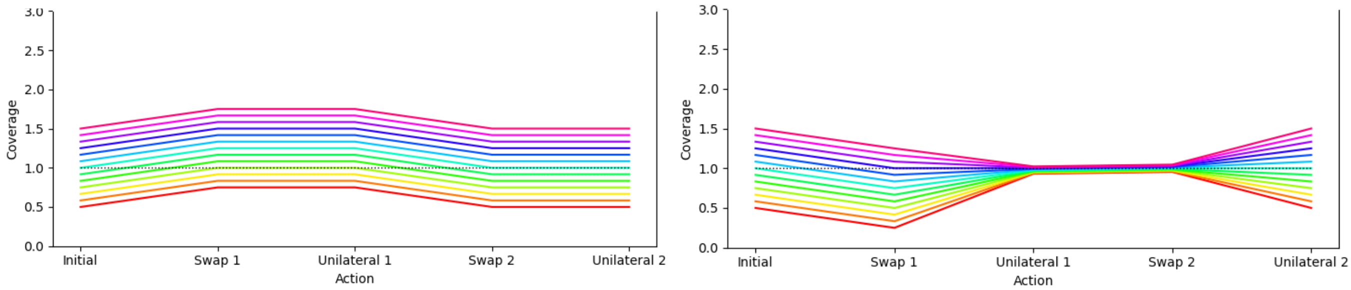

Figure 5 : Coverage evolution for Type 2 routes across multiple generic exchange configurations
Type 2: Swap $T_1 \to T_2$, deposit T_2, reverse swap $T_2 \to T_1$, withdraw T_2

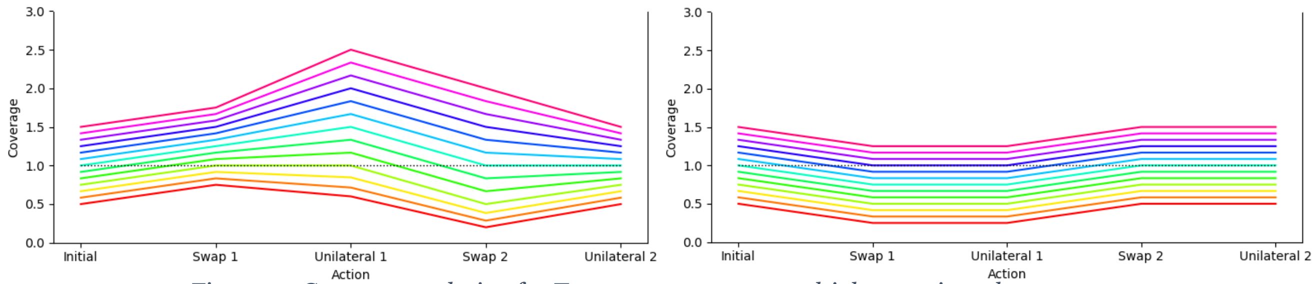

Figure 5 : Coverage evolution for Type 3 routes across multiple generic exchange configurations
Type 3: Swap $T_1 \to T_2$, withdraw T_1, reverse swap $T_2 \to T_1$, deposit T_1

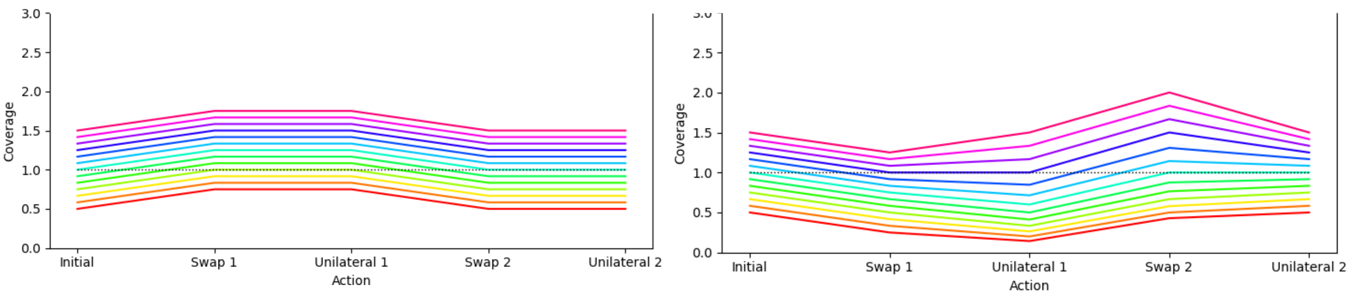

Figure 5 : Coverage evolution for Type 4 routes across multiple generic exchange configurations
Type 4: Swap $T_1 \to T_2$, withdraw T_2, reverse swap $T_2 \to T_1$, deposit T_2

Quantitative Example

We focus on a specific route to work through in algebraic detail in order to derive a full model of arbitrage across other arbitrage route types, as well as highlight errors in the Platypus model.

We will explore a Type 1 route, as it performs a deposit before a withdrawal and, heuristically, has the maximum potential (this will be justified shortly).

- Swap: T_i into T_j: $\left(r_i^{(0)}, r_j^{(0)}\right) \to \left(r_i^{(1)}, r_j^{(1)}\right)$ where $r_i^{(1)} > r_i^{(0)}$ and $r_j^{(1)} < r_j^{(0)}$
- Unilateral: Deposit a quantity D of T_i, $r_i^{(1)} \to r_i^{(2)}$ (with $r_j^{(2)} = r_j^{(1)}$)
- Swap T_1 into T_i: $\left(r_i^{(2)}, r_j^{(2)}\right) \to \left(r_i^{(3)}, r_j^{(3)}\right)$ where $r_i^{(3)} < r_i^{(2)}$ and $r_j^{(3)} > r_j^{(2)}$
- Reverse unilateral: Withdraw a quantity W of T_i, $r_i^{(3)} \to r_i^{(4)}$ (with $r_j^{(4)} = r_j^{(3)}$)

The initial configuration of the pool is

- Token i: $A_i^{(0)}, L_i^{(0)}, r_i^{(0)}$
- Token j: $A_j^{(0)}, L_j^{(0)}, r_j^{(0)}$

In addition to tracking the evolution of the pool's status we will track the (net) holdings of a trader trying to engage in arbitrage.

The trader has 4 quantities we take:

- Tokens T_i , T_j holdings: $X_i T_i$ and $X_j T_j$
- Platypus minted tokens[7], C_i , C_j , able to withdraw T_i , T_j deposits: $Y_i T_i$ and $Y_j T_j$

Since we only care about the net change we set $X_i^{(0)} = X_j^{(0)} = Y_i^{(0)} = Y_j^{(0)} = 0$.

Initial Swap

The trader sells $\Delta_i^{(1)} T_i$ to the pool. We calculate slippage using to determine the terminal exchange rate.

- Pool In:
  - Amount: $\Delta_i^{(1)}$ of T_i
  - Assets: $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - Liability: $L_i^{(1)} = L_i^{(0)}$

[7] We do not actually need to distinguish between deposits of different liquidity tokens, only one token type is deposited/withdrawn, and the liquidity provider just needs to be able to withdraw their deposit, but the analysis is more symmetric, clear and generalises to other arbitrage routes.

- ○ Cost-less coverage: $r_i^{(1)} = \frac{A_i^{(1)}}{L_i^{(1)}} = r_i^{(0)} + \frac{\Delta_i^{(1)}}{L_i^{(0)}}$
- ○ Marginal slippage: $S_i^{(1)} = \frac{g\left(r_i^{(1)}\right)-g\left(r_i^{(0)}\right)}{r_i^{(1)}-r_i^{(0)}} = \frac{L_i^{(0)}}{\Delta_i^{(1)}}\left(g\left(r_i^{(1)}\right) - g\left(r_i^{(0)}\right)\right)$

- Pool Out:
  - ○ Amount: $f_{i\to j}\Delta_i^{(1)}$ of T_j
  - ○ Assets: $A_j^{(1)} = A_j^{(0)} - f_{i\to j}\Delta_i^{(1)}$
  - ○ Liability: $L_j^{(1)} = L_j^{(0)}$
  - ○ Cost-less coverage: $r_j^{(1)} = \frac{A_j^{(1)}}{L_j^{(1)}} = r_j^{(0)} - f_{i\to j}\frac{\Delta_i^{(1)}}{L_j^{(0)}}$
  - ○ Marginal slippage: $S_j^{(1)} = \frac{g\left(r_j^{(1)}\right)-g\left(r_j^{(0)}\right)}{r_j^{(1)}-r_j^{(0)}} = -\frac{L_j^{(0)}}{f_{i\to j}\,\Delta_i^{(1)}}\left(g\left(r_j^{(1)}\right) - g\left(r_j^{(0)}\right)\right)$

Generate full slippage and exchange rates:

- Slippage $S_{i\to j}^{(1)} = S_i^{(1)} - S_j^{(1)} = \frac{L_i^{(0)}}{\Delta_i^{(1)}}\left(g\left(r_i^{(1)}\right) - g\left(r_i^{(0)}\right)\right) + \frac{L_j^{(0)}}{f_{i\to j}\,\Delta_i^{(1)}}\left(g\left(r_j^{(1)}\right) - g\left(r_j^{(0)}\right)\right)$
- Terminal Exchange Rate $f_{i\to j}^{*} = f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)$

The actual swap using the terminal exchange rate is then performed:

- Pool In:
  - ○ Amount: $\Delta_i^{(1)}$ of T_i
  - ○ Assets: $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - ○ Liability: $L_i^{(1)} = L_i^{(0)}$
  - ○ Coverage: $r_i^{(1)} = \frac{A_i^{(1)}}{L_i^{(1)}} = r_i^{(0)} + \frac{\Delta_i^{(1)}}{L_i^{(0)}}$
- Pool Out:
  - ○ Amount: $f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$ of T_j
  - ○ Assets: $A_j^{(1)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$
  - ○ Liability: $L_j^{(1)} = L_j^{(0)}$
  - ○ Coverage: $r_j^{(1)} = \frac{A_j^{(1)}}{L_j^{(1)}} = r_j^{(0)} - \frac{f_{i\to j}\left(1-S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}}{L_j^{(0)}}$

The trader's assets change in the opposite manner:

- Trader out = Pool in $X_i^{(1)} = X_i^{(0)} - dA_i^{(0)} = -\Delta_i^{(1)}$
- Trader in = Pool out $X_j^{(1)} = X_j^{(0)} - dA_j^{(0)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$

Deposit

The trader deposits $D\ T_i$ to the pool, which could experience a penalty of β if $r_i^{(1)} > 1$.

- Pool In:
  - Amount: D of T_i
  - Assets: $A_i^{(2)} = A_i^{(1)} + D$
  - Liability: $L_i^{(2)} = L_i^{(1)} + (D - \beta)$
  - Coverage: $r_i^{(2)} = \frac{A_i^{(1)} + D}{L_i^{(1)} + (D - \beta)}$

To "compensate" the trader the pool will mint a set of smart tokens corresponding to the change in liability.

- Pool Out:
  - Amount: $D - \beta$ of C_i

The trader's assets change in the opposite manner:

- Trader out = Pool in: $X_i^{(2)} = X_i^{(1)} - D$
- Trader in = Pool out: $Y_i^{(2)} = Y_i^{(1)} + (D - \beta) = D - \beta$

All other quantities for the trader and pool are unchanged.

Reverse Swap

A second swap, in the opposite direction, is then performed by the trader, selling $\Delta_j^{(2)}\ T_j$ to the pool.

- Pool In:
  - Amount: $\Delta_j^{(2)}$ of T_j
  - Assets: $A_j^{(2)} \to A_j^{(3)} = A_j^{(2)} + \Delta_j^{(2)}$
  - Liability: $L_j^{(2)} \to L_j^{(3)} = L_j^{(2)}$
  - Coverage: $r_j^{(2)} \to r_j^{(3)} = \frac{A_j^{(3)}}{L_j^{(3)}} = r_j^{(2)} + \frac{\Delta_j^{(2)}}{L_j^{(2)}}$
  - Marginal slippage: $S_j^{(2)} = \frac{g\left(r_j^{(3)}\right) - g\left(r_j^{(2)}\right)}{r_j^{(3)} - r_j^{(2)}} = \frac{L_j^{(2)}}{\Delta_j^{(2)}}\left(g\left(r_j^{(3)}\right) - g\left(r_j^{(2)}\right)\right)$
- Pool Out:
  - Amount: $f_{j\to i}\Delta_j^{(2)}$ of T_i
  - Assets: $A_i^{(2)} \to A_i^{(3)} = A_i^{(2)} - f_{j\to i}\Delta_1^{(2)}$

- Liability: $L_i^{(2)} \to L_i^{(3)} = L_i^{(2)}$
- Coverage: $r_i^{(2)} \to r_i^{(3)} = \frac{A_i^{(3)}}{L_i^{(3)}} = r_i^{(2)} - \frac{f_{j\to i}\Delta_1^{(2)}}{L_i^{(2)}}$
- Marginal slippage: $S_i^{(1)} = \frac{g(r_i^{(1)})-g(r_i^{(0)})}{r_i^{(1)}-r_i^{(0)}} = -\frac{L_i^{(2)}}{f_{j\to i}\Delta_1^{(2)}}\left(g(r_i^{(3)}) - g(r_i^{(2)})\right)$

Generate full slippage and exchange rates:

- Slippage $S_{j\to i}^{(2)} = S_j^{(2)} - S_i^{(2)} = \frac{L_j^{(2)}}{\Delta_j^{(2)}}\left(g(r_j^{(3)}) - g(r_j^{(2)})\right) + \frac{L_i^{(2)}}{f_{j\to i}\,\Delta_j^{(2)}}\left(g(r_i^{(3)}) - g(r_i^{(2)})\right)$
- Terminal Exchange Rate $f_{j\to i}^* = f_{j\to i}\left(1 - S_{j\to i}^{(1)}\right)(1-h)$

The actual swap using the terminal exchange rate is then performed:

- Pool In:
  - Amount: $\Delta_j^{(2)}$ of T_j
  - Assets: $A_j^{(3)} = A_j^{(2)} + \Delta_j^{(2)}$
  - Liability: $L_j^{(3)} = L_j^{(2)}$
  - Coverage: $r_j^{(3)} = \frac{A_j^{(3)}}{L_j^{(3)}} = r_j^{(2)} + \frac{\Delta_j^{(2)}}{L_j^{(2)}}$
- Pool Out:
  - Amount: $f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\Delta_j^{(2)}$ of T_i
  - Assets: $A_i^{(3)} = A_i^{(2)} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\Delta_j^{(2)}$
  - Liability: $L_i^{(3)} = L_i^{(2)}$
  - Coverage: $r_i^{(3)} = \frac{A_i^{(3)}}{L_i^{(3)}} = r_i^{(2)} - \frac{f_{j\to i}\left(1-S_{j\to i}^{(2)}\right)(1-h)\Delta_j^{(2)}}{L_i^{(2)}}$

The trader's assets change in the opposite manner:

- Trader out = Pool in $X_j^{(3)} = X_j^{(2)} - dA_j^{(2)} = X_j^{(2)} - \Delta_j^{(2)}$
- Trader in = Pool out $X_i^{(3)} = X_i^{(2)} - dA_i^{(2)} = X_i^{(2)} + f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\Delta_j^{(2)}$

Withdrawal

The trader withdraws their deposited liquidity by redeeming *all* Platypus tokens in their portfolio, totalling $Y_i^{(3)}C_i = (D - \beta)C_i$, for T_i, potentially paying a penalty α if $r_i^{(3)} < 1$.

- Pool Out:
  - Amount: $(D - \beta) - \alpha$ of T_i

- Assets: $A_i^{(4)} = A_i^{(3)} - \big((D-\beta)-\alpha\big)$
- Liability: $L_i^{(4)} = L_i^{(3)} - (D-\beta)$
- Coverage: $r_i^{(4)} = \frac{A_i^{(3)} - ((D-\beta)-\alpha)}{L_i^{(3)} - (D-\beta)}$

All other pool quantities are unchanged. The trader's portfolio will change in the opposite manner:

- Trader out = Pool in: $Y_i^{(4)} = Y_i^{(3)} - (D-\beta) = 0$
- Trader in = Pool out: $X_i^{(4)} = X_i^{(3)} + \big((D-\beta)-\alpha\big)$

All other quantities for the trader and pool are unchanged.

Evolution

Using the above results, we summarise the evolution of the pool assets and liabilities in terms of its initial state and the 4 actions.

- Pool T_i values
  - $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - $A_i^{(2)} = A_i^{(0)} + \Delta_i^{(1)} + D$
  - $A_i^{(3)} = A_i^{(0)} + \Delta_i^{(1)} + D - f_{j\to i}\big(1 - S_{j\to i}^{(2)}\big)(1-h)\,\Delta_j^{(2)}$
  - $A_i^{(4)} = A_i^{(0)} + \Delta_i^{(1)} + D - f_{j\to i}\big(1 - S_{j\to i}^{(2)}\big)(1-h)\,\Delta_j^{(2)} - \big((D-\beta)-\alpha\big)$
  - $L_i^{(1)} = L_i^{(0)}$
  - $L_i^{(2)} = L_i^{(1)} + (D-\beta)$
  - $L_i^{(3)} = L_i^{(2)} + (D-\beta)$
  - $L_i^{(4)} = L_i^{(2)}$
- Pool T_j values
  - $A_j^{(1)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)}$
  - $A_j^{(2)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)}$
  - $A_j^{(3)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} + \Delta_j^{(2)}$
  - $A_j^{(4)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} + \Delta_j^{(2)}$
  - $L_j^{(1)} = L_j^{(0)}$
  - $L_j^{(2)} = L_j^{(0)}$
  - $L_j^{(3)} = L_j^{(0)}$
  - $L_j^{(4)} = L_j^{(0)}$

The portfolio's end state has zero $C_{1,2}$ positions so we need only consider its $T_{1,2}$ holdings.

- Portfolio T_1 values
  - $X_j^{(1)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\,\Delta_i^{(1)}$
  - $X_j^{(2)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\,\Delta_i^{(1)}$
  - $X_j^{(3)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\,\Delta_i^{(1)} - \Delta_j^{(2)}$
  - $X_j^{(4)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\,\Delta_i^{(1)} - \Delta_j^{(2)}$
- Portfolio T_i values
  - $X_i^{(1)} = -\Delta_i^{(1)}$
  - $X_i^{(2)} = -\Delta_i^{(1)} - D$
  - $X_i^{(3)} = -\Delta_i^{(1)} - D + f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$
  - $X_i^{(4)} = -\Delta_i^{(1)} - D + f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)} + \left((D-\beta)-\alpha\right)$

The total net change, in T_i units[8], is a combination of the portfolio components:

$$V\left(X_i, X_j \middle| T_i\right) = X_i + f_{j\to i}X_j$$

Using $f_{i\to j}f_{j\to i} = 1$ we can calculate this for the final portfolio:

$$V = \left(-\Delta_i^{(1)} + f_{j\to i}\Delta_j^{(2)}\left(1 - S_{j\to i}^{(2)}\right)(1-h) - (\alpha+\beta)\right) + f_{j\to i}\left(f_{i\to j}\Delta_i^{(1)}\left(1 - S_{i\to j}^{(1)}\right)(1-h) - \Delta_j^{(2)}\right)$$

Simplifying and collecting terms:

$$V = \Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)(1-h) - 1\right] + f_{j\to i}\Delta_j^{(2)}\left[\left(1 - S_{j\to i}^{(2)}\right)(1-h) - 1\right] - (\alpha+\beta)$$

This expression makes it easy to identify the source of each term:

- Swap 1: $+\Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)(1-h) - 1\right]$
- Deposit: $-\beta$
- Swap 2: $+f_{j\to i}\Delta_j^{(2)}\left[\left(1 - S_{j\to i}^{(2)}\right)(1-h) - 1\right]$
- Withdrawal $-\alpha$

This demonstrates that, for this type of arbitrage route, the withdrawal step can never aid in arbitrage. The impact of the deposit is less trivial, as it influences $S_{i\to 1}^{(1)}$ through its modification of the r_i coverage ratio.

[8] A net change of +10 is equivalent to a gain equal in value to 10 T_i.

Swap Constraint for Arbitrage

The above results are true for arbitrary swaps $T_i \to T_j$ followed by $T_j \to T_i$. To identify an arbitrage procedure, we want to "close the loop", zeroing out the net change in either the portfolio's T_i holdings or its T_j holdings, just as was done for the Platypus token C_i holdings.

Unlike the net T_i holdings, the net T_j holdings do not depend on the deposit/withdrawal penalties, so we choose this option.

$$X_j^{(4)} = 0 \quad \Rightarrow \quad \Delta_j^{(2)} = f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$$

This demonstrates that the second swap depends on the first swap *and* its slippage. Inserting this into the portfolio's net change and simplifying:

$$V = \Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)(1-h) - 1\right] + \Delta_i^{(1)}\left[\left(1 - S_{j\to i}^{(2)}\right)(1-h) - 1\right]\left(1 - S_{i\to j}^{(1)}\right)(1-h) - (\alpha + \beta)$$

At this point we highlight that the first swap's contribution is *linear* in slippage, but the second swap's contribution is now *quadratic* in slippage due to the slippage dependence of $\Delta_j^{(2)}$.

Collecting terms of $1 - h$ gives a form that makes the impact of fees clear:

$$V = \Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta) = (1-h)^2 V_{\text{slip}} + V_{-}$$

The slippage dependent contribution to V, V_{slip}, is the only *potential* source of positive value. The remaining contribution, $V_{-} = -\Delta_2^{(1)} - \alpha - \beta$, is strictly negative.

Arbitrage protection is then implemented by designing α, β such that V is *never* positive. From the above result, and the h dependence of $S_{j\to i}^{(2)}$ via nested slippage, it follows that if arbitrage is impossible for $h = 0$ it will be impossible for all $h > 0$, allowing a simplified analysis.

It is important to acknowledge that α, β must preclude $V > 0$ for *any* combinations of swaps, deposits and withdrawals and thus Types 2, 3 and 4 arbitrage routes. We could repeat this full analysis for Types 2, 3 and 4 to extract a general result but this is considerable, largely duplicated, effort.

Fortunately, considerable effort can be avoided[9] with a modicum of thought – the structure of V clearly indicates the general form of the potential arbitrage for the generic sequence: swap $T_i \to T_j$, unilateral on T_k, swap $T_j \to T_i$, reverse unilateral on T_k, when expressed in units of T_k, is:

$$V = \Delta_m^{(n)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$$

This is the *Arbitrage Value Equation (AVE)*.

[9] If considerable effort is preferred, then the Appendix section *Alternative Arbitrage Routes* derives the general result for all inequivalent arbitrage routes.

We use T_k units because the deposit and withdrawal penalties are on T_k. The impact of orderings, swap directions, unilateral actions and amounts are purely through the evolution of the assets, liabilities, and coverages, from which the slippages and penalties can be calculated and plugged into the AVE.

This breaks down into 3 parts, categorised by their slippage orders:

- Quadratic: $V_2 = +\Delta_m^{(n)}(1-h)^2\left(S_{i\to j}^{(1)} S_{j\to i}^{(2)}\right)$
- Linear: $V_1 = -\Delta_m^{(n)}(1-h)^2\left(S_{i\to j}^{(1)} + S_{j\to i}^{(2)}\right)$
- Constant: $V_0 = +\Delta_m^{(n)}[(1-h)^2 - 1] - (\alpha + \beta)$

It is important to note that for instances of double negative slippage, $S_{i\to j}^{(1)} < 0$ and $S_{j\to i}^{(2)} < 0$, the linear *and* quadratic terms give positive contributions, potentially increasing arbitrage effects.

The associated "loop closing" swap constraint is similarly deducible:

$$\Delta_{m\prime}^{(n\prime)} = f_{i\to j}\left(1 - S_{i\to j}^{(n)}\right)(1-h)\Delta_m^{(n)}$$

This is the *Arbitrage Swap Constraint (ASC)*.

The specific expressions for each of the 4 arbitrage route types are given in the Appendix.

We will return to exploring the relative importance of these terms when we construct a new arbitrage protection.

Platypus Arbitrage

Stated Values

We are now able to compare with Platypus's analysis, specifically Section 5.1 – "Deposit Arbitrage, Arbitrage Procedure". It outlines a 3-step arbitrage sequence[10] with the following properties:

1. r_j has its "loop closed", returning to its initial state
2. The second swap returns exactly the T_i output by the first swap

Property 1 is explicitly stated in the "Swap and Deposit path":

$$(r_i, r_j) \xrightarrow{\text{swap}} (r_i', r_j') \xrightarrow{deposit} (r_i'', r_j') \xrightarrow{\text{swap}} (r_i^*, r_j)$$

[10] The 4th step, withdrawal of the deposited amount, is not included but it can only hinder arbitrage.

Property 2 can be seen by the lack of swap size dependence in $r_i^* = \frac{A_i+D}{L_i+D}$, where its initial state had been $r_i = \frac{A_i}{L_i}$. It can also be seen in the net slippage expression in the Yellow Paper, V_{YP}:

$$V_{YP} = -yS_{j\to i} + f_{j\to i}(-yf_{i\to j})S'_{i\to j} = -y(S_{j\to i} + S'_{i\to j})$$

In notation previously developed for our quantitative analysis:

1. $r_j^{(4)} = r_j^{(0)}$
2. $\Delta_j^{(1)} = f_{i\to j}\Delta_i^{(2)}$ or equivalently $\Delta_i^{(2)} = f_{j\to i}\Delta_j^{(1)}$

Property 2 is the *Platypus Swap Constraint (PSC)*.

Contradictions

The two stated properties in the Platypus methodology are *contradictory*.

To align with our notation, we note this is a $T_j \to T_i$ swap followed by a T_i deposit. This is the Type 2 arbitrage route under the token index swap $i \leftrightarrow j$. To match the Yellow Paper, we also turn fees/penalties off in our notation, $h = \alpha = \beta = 0$ and express the portfolio net gain in terms of T_j.

Applying these changes/restrictions to the relevant Type 2 expressions in the Appendix:

- $A_j^{(4)} = A_j^{(0)} + \Delta_j^{(1)} - f_{i\to j}(1 - S_{i\to j}^{(2)})\,\Delta_i^{(2)}$
- $V = -\Delta_j^{(1)} S_{j\to i}^{(1)} - f_{i\to j}\Delta_i^{(2)} S_{i\to j}^{(2)}$

Applying the PSC, $\Delta_i^{(2)} = f_{j\to i}\Delta_j^{(1)}$, to the portfolio net gain:

$$V \to V_{Plat} \equiv -\Delta_j^{(1)}(S_{j\to i}^{(1)} + S_{i\to j}^{(2)}) \quad \sim \quad -y(S_{j\to i} + S'_{i\to j})$$

We have been able to replicate the Platypus Yellow Paper total slippage result. Unfortunately, it does not match the general AVE expression previously derived because it lacks the quadratic term, the slippage of slippage:

$$V_2 = \Delta_j^{(1)}(S_{j\to i}^{(1)} S_{i\to j}^{(2)}).$$

Furthermore, applying the PSC to the T_j assets does not reproduce the Yellow Paper value:

$$A_j^{(4)} = A_j^{(0)} + \Delta_j^{(1)} - f_{i\to j}(1 - S_{i\to j}^{(2)})\,\Delta_i^{(2)} \to A_j^{(0)} + S_{i\to j}^{(2)}\,\Delta_j^{(1)} \neq A_j^{(0)}$$

Therefore, the PSC has *not* closed the T_j loop because it does not account for the *second* slippage in the assets seen in the ASC.

Similarly, if we apply the PSC to the T_i assets we do not recover the r_i^* expression used in Platypus, again due to a slippage term:

$$A_i^{(3)} = A_i^{(0)} - f_{j\to i}\big(1 - S_{j\to i}^{(1)}\big)\,\Delta_j^{(1)} + D + \Delta_i^{(2)} \quad \to \quad A_i^{(0)} + S_{j\to i}^{(1)}\,\Delta_i^{(2)} + D \;\neq A_i^{(0)} + D$$

Therefore, under the PSC the post-reverse-swap state of r_i, $r_i^{(3)}$, also includes a slippage term:

$$r_i^{(3)} = \frac{A_i^{(0)} + S_{j\to i}^{(1)}\,\Delta_i^{(2)} + D}{L_i^{(2)} + D} = \frac{A_i^{(0)} + D}{L_i^{(2)} + D} + \frac{S_{j\to i}^{(1)}\,\Delta_i^{(2)}}{L_i^{(2)} + D} = r_i^* + \frac{\Delta_i^{(2)} S_{j\to i}^{(1)}}{L_i^{(2)} + D}$$

Compare this to applying the ASC, $\Delta_j^{(1)} = f_{i\to j}\big(1 - S_{i\to j}^{(2)}\big)\,\Delta_i^{(2)}$, which closes the T_j loop:

$$A_i^{(3)} = A_i^{(0)} - f_{j\to i}\big(1 - S_{j\to i}^{(1)}\big)\Delta_j^{(1)} + D + \Delta_i^{(2)} \quad \to \quad A_i^{(0)} + D - \Delta_i^{(2)}\big[\big(1 - S_{j\to i}^{(1)}\big)\big(1 - S_{i\to j}^{(2)}\big) - 1\big]$$

Therefore, the r_i coverage after the T_j loop closing ASC includes not one but *two* slippage factors:

$$r_i^{(3)} = \frac{A_i^{(3)}}{L_i^{(3)}} = r_i^* + \frac{\Delta_i^{(2)}\big[1 - \big(1 - S_{j\to i}^{(1)}\big)\big(1 - S_{i\to j}^{(2)}\big)\big]}{L_i^{(0)} + D}$$

The *true* discrepancy in $A_i^{(3)}$ precisely corresponds to the value change in the exchange due to the arbitrage route, as seen in the AVE, showing an overall "conservation of value" between the exchange and the trader.

Rather than being due to a more fundamental arbitrage prevention scheme Platypus's use of $r_i^{(3)} = r_i^*$ is only consistent with an arbitrage-free system because it is consistent with a zero-slippage system, which is arbitrage-free by default. In a system with slippage the expressions used in the Yellow Paper's arbitrage analysis are not consistent with each other or "conservation of value" between exchange and trader.

More generally, throughout Sections 4 & 5 of the Platypus Yellow Paper expressions consistent with the PSC, $\Delta_j^{(1)} = f_{i\to j}\,\Delta_i^{(2)}$, are mixed with expressions consistent with the ASC, $\Delta_j^{(1)} = f_{i\to j}\big(1 - S_{i\to j}^{(2)}\big)\,\Delta_i^{(2)}$. While it leads to several elegant looking results, particularly those based on the convexity of $g(r)$, all the conclusions stemming from this constraint are, at best, unjustified and, at worst, invalid.

Non-Independence

In addition to deriving a net gain expression that is only linear, not quadratic, in slippage Platypus the $r_i^{(3)} \neq r_i^*$ and lack of T_j closure produces a further error in Platypus's arbitrage expression.

Recalling the form of V_{Plat} we can expand it out in terms of marginal slippages:

$$V_{Plat} = -\Delta_j^{(1)}\big(S_{j\to i}^{(1)} + S_{i\to j}^{(2)}\big) = -\Delta_j^{(1)}\big[S_j^{(1)} - S_i^{(1)} + S_i^{(2)} - S_j^{(2)}\big]$$

Expanding the definition of the marginals in terms of the individual coverage ratios:

$$V_{Plat} = -\Delta_j^{(1)}\left[S\left(r_j^{(0)}, r_j^{(1)}\right) - S\left(r_i^{(0)}, r_i^{(1)}\right) + S\left(r_i^{(2)}, r_i^{(3)}\right) - S\left(r_j^{(2)}, r_j^{(3)}\right)\right]$$

Collecting $r_{i,j}$ dependencies, accounting for S being a symmetric function and using $r_i^{(2)} = r_i^{(1)}$:

$$V_{Plat} = -\Delta_j^{(1)}\left[S\left(r_j^{(0)}, r_j^{(1)}\right) - S\left(r_j^{(2)}, r_j^{(3)}\right)\right] - \Delta_j^{(1)}\left[S\left(r_i^{(3)}, r_i^{(1)}\right) - S\left(r_i^{(0)}, r_i^{(1)}\right)\right]$$

This is the full expression for net value gained by a trader doing a *Type 1* arbitrate route where the reverse swap is the PSC, which amounts to neglecting slippage.

This *almost* matches the general structure of expression given in the Yellow Paper, V_{YP}:

$$V_{YP} = -y\left[\left(S_j - S_i\right) + \left(S_i' - S_j'\right)\right] = -y\left[\left(S_j - S_j'\right) + \left(S_i' - S_i\right)\right] \to -y\left[\left(S_j - S_j\right) + \left(S_i' - S_i\right)\right]$$

The mismatch arises from the fact the Yellow Paper uses $r_j^{(0)} = r_j^{(3)}$, causing the T_j dependence to cancel out from the extracted value.

$$V_{YP} = V_{YP,j} + V_{YP,i} \ \to V_{YP,i}$$

But the PSC does not close the T_j loop and $S_j^{(1)} \neq S_j^{(2)}$. Without this loop closure it is not correct to reduce V_{YP} down further by saying $V_{YP,j} = 0$ and without this reduction we can see V_{YP} is dependent on r_i **and** r_j.

Discontinuous Penalties

Setting aside this issue, we continue following the Yellow Paper's derivation of arbitrage protection by dropping $V_{YP,j}$ and putting $V_{YP,i}$ into our notation:

$$V_{YP}(\alpha, \beta) = +\Delta_j^{(1)}\left[S\left(r_i^{(0)}, r_i^{(1)}\right) - S\left(r_i^{(2)}, r_i^{(3)}\right)\right] - (\alpha + \beta) = V_{YP}(0{,}0) - (\alpha + \beta)$$

This is the same expression regardless of whether the order is "withdrawal then deposit" or "deposit then withdrawal", as we have proven in this article, as well as the Platypus Yellow Paper demonstrates (Platypus Team, 2022, pp. 20, 23), only the specific values for assets, liabilities and coverages change.

As previously stated, the challenge of arbitrage protection is to then design α, β to prevent $V_{YP}(\alpha, \beta)$ *ever* going positive. Platypus design *univariate* expressions for α, β by identifying the maximum values of $V_{YP}(0{,}0)$ in different arbitrage routes.

Recalling the expressions for the penalties from the start of the article, but now in our notation:

$$\alpha\left(W \middle| r_j^{(1)}, L_j^{(1)}\right) = \mathrm{Ind}\left(r_j^{(1)} < 1\right) \underbrace{\left[g(r')\left(L_j^{(1)} - W\right) - g\left(r_j^{(1)}\right)L_j^{(1)} + g(1)W\right]}_{\omega\left(W \middle| r_j^{(1)}, L_j^{(1)}\right)} \quad , \quad r' = \frac{r_j^{(1)}L_j^{(1)} - W}{L_j^{(1)} - W}$$

$$\beta\left(D \middle| r_j^{(1)}, L_j^{(1)}\right) = \mathrm{Ind}\left(r_j^{(1)} > 1\right) \underbrace{\left[g(r')\left(L_j^{(1)} + D\right) - g\left(r_j^{(1)}\right)L_j^{(1)}\right]}_{\delta\left(D \middle| r_j^{(1)}, L_j^{(1)}\right)} \quad , \quad r' = \frac{r_j^{(1)}L_j^{(1)} + D}{L_j^{(1)} + D}$$

Immediately we can identify an issue – the expressions for r' are incorrect as they do not include the quadratic slippage term identified in the previous section and the convenient parameterisation the slip-less case provides is pivotal to the maximisation procedures used in the Yellow Paper (Platypus Team, 2022, pp. 22, 25)

Given the multitude of issues identified with the PSC and incorrect estimate of net gain, V_{YP}, it follows that the functions $\omega(W|r, L)$ and $\delta(D|r, L)$ are, at best, untrustworthy, and at worst inaccurate.

A second issue arises from the other factor in the penalty expressions – the Indictor Function. Each penalty has a domain of applicability – $r > 1$ for deposit penalties, $r < 1$ for withdrawals – where each action decreases the coverage.

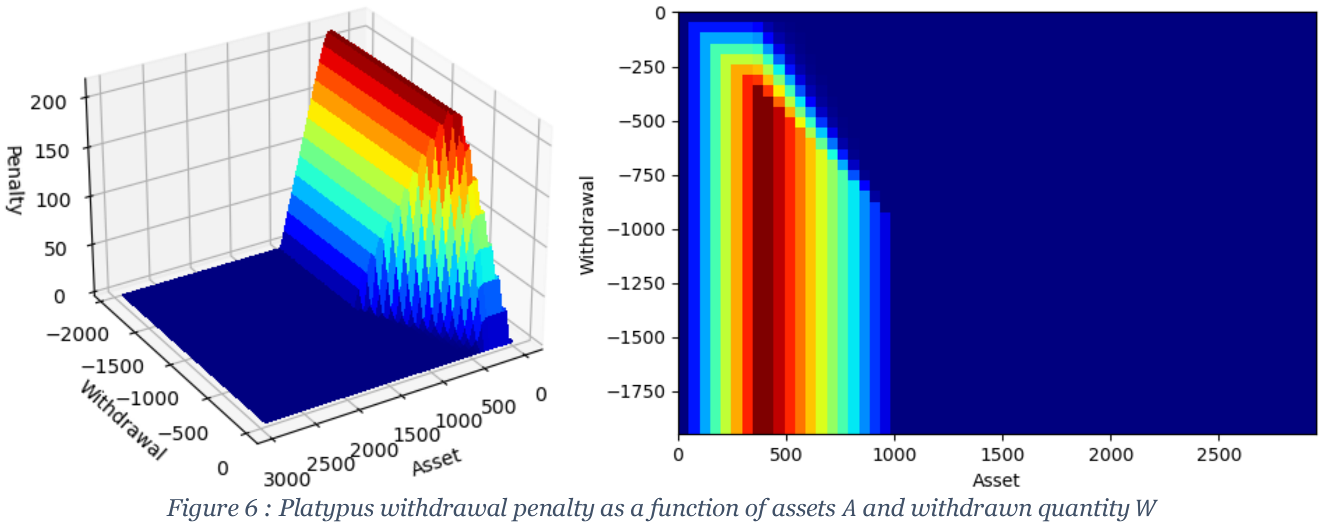

Figure 6 : Platypus withdrawal penalty as a function of assets A and withdrawn quantity W (Liability fixed at $L = 1000$). Note W is capped to $W \rightarrow -min(-W, 0.999A, 0.999L)$ to prevent pool insolvency.

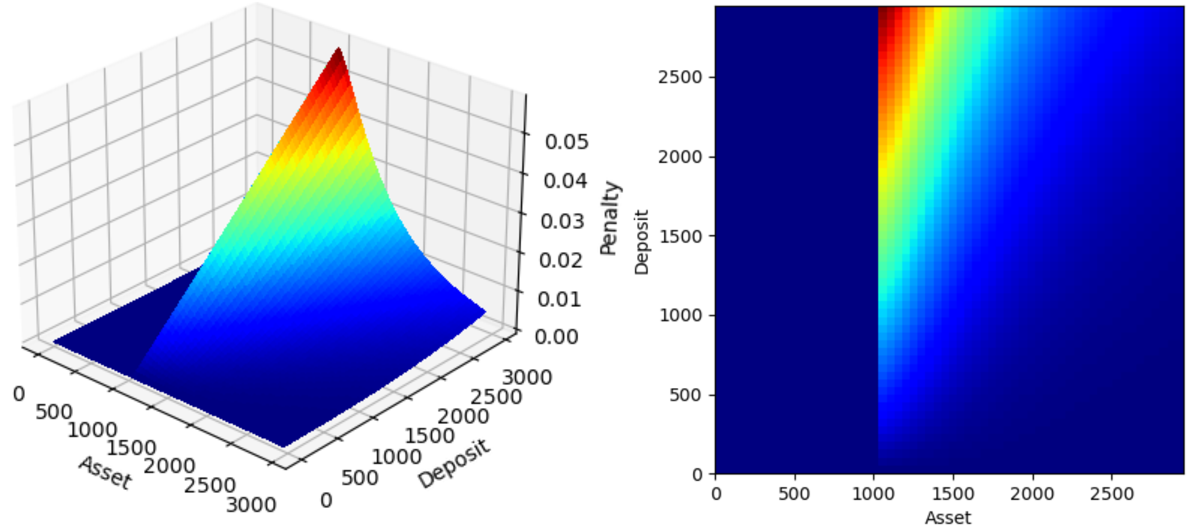

Figure 8 : Platypus deposit penalty as a function of assets A and withdrawn quantity W (Liability fixed at L=1000)

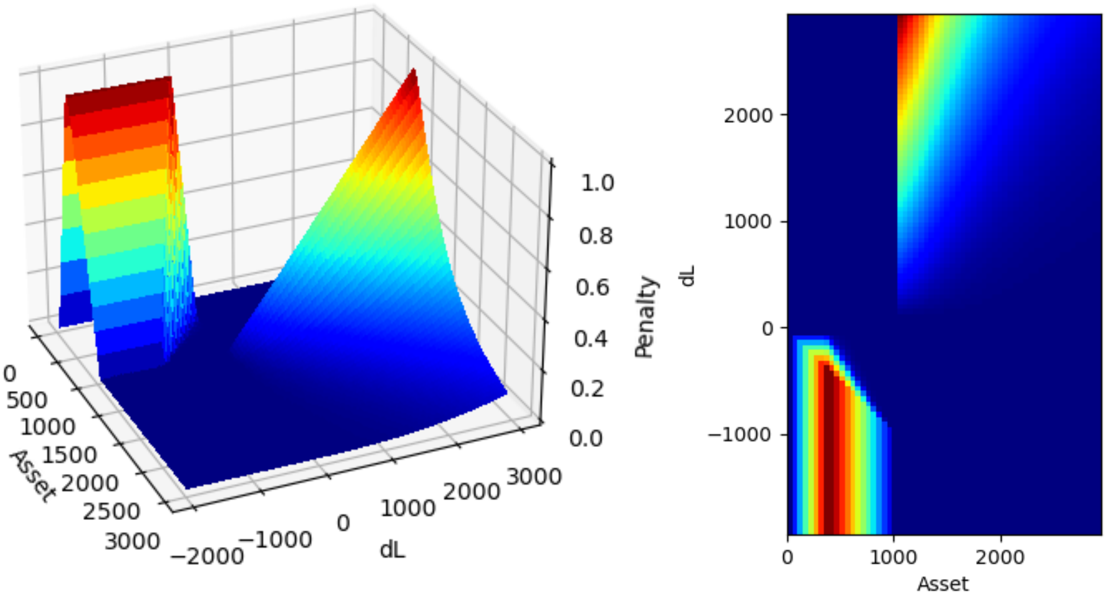

Figure 7 : Platypus combined withdrawal and deposit penalty as a function of assets A and withdrawn quantity W (Liability fixed at L=1000). Due to significantly different scales the withdrawal penalty is scaled down by a factor of 213.96 and the deposit penalty scaled up by a factor of 17.2 to give each a local maximum of 1. The relative scale difference is $213.96 \times 17.2 \approx 3680$.

To give qualitative insight into the issue with the Indicator Functions we generate penalty surfaces for withdrawals (Figure 6) and deposits (Figure 8) and their relative positions/profiles (Figure 7). Immediately an issue with the deposit profile is apparent – the increasingly large discontinuity at $r = 1$. This contrasts with the withdrawal profile, which smoothly tends to 0 as $r \rightarrow 1$.

As we will shortly discuss, the use of indicator functions is not only unjustified but provides an arbitrage route.

Swap Calibration

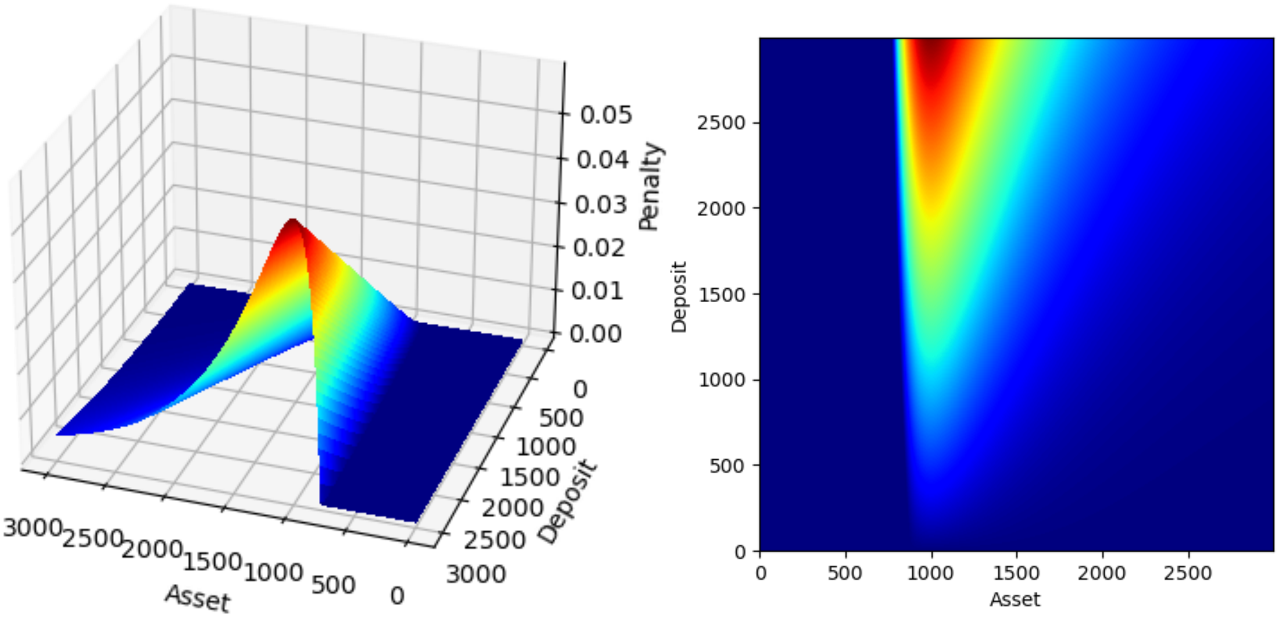

Figure 9 : Deposit penalty function, without the $r > 1$ indicator function, for all coverage values, showing positive value in $r < 1$ region.

In the analysis of arbitrage (Platypus Team, 2022, pp. 20, 23) the Yellow Paper considers a generic swap $T_j \to T_i$ of quantity y followed by the unilateral action. In both the deposit and withdrawal cases the analysis states "The new coverage ratio decreases ...". The Yellow Paper deposit arbitrage procedure is a Type 2 route, while the withdrawal deposit procedure is a Type 4 route.

We previously noted, using the set of scenarios in Figure 5 (Type 2) and Figure 5 (Type 4), that this is *not* the generic behaviour for deposits or withdrawals. In general deposits push coverage towards 1 and withdrawals push coverage away from 1.

Therefore, swaps that cross the $r = 1$ line flip the coverage impact of the unilateral action. The reverse swaps will then, aside from some "edge cases" where slippage is comparable to $|L - A|$, cross back over the $r = 1$ line again.

This is important because the Indicator Support Functions $\mathrm{Ind}\left(r_j^{(1)}>1\right)$, $\mathrm{Ind}\left(r_j^{(1)}<1\right)$ introduce unjustified discontinuities in the penalties.

This can be seen from the expression for $\delta(D|r, L)$. Suppose $r = 1 - \eta$ for $|\eta| \ll 1$ – if $\eta > 0$ the coverage is below 1 and if $\eta < 0$ the coverage is above 1. We then perform a large deposit, moving the coverage closer to $r = 1$, equivalently $r \to r' = \frac{rL+D}{L+D} = 1 - \xi\eta$ where $0 < \xi < 1$ is a monotonically decreasing function of D ($\xi \to 0$ as $D \to \infty$).

$$\delta(D|r,L) = g(1-\xi\,\eta)(L+D) - g(1-\eta)L \quad , \quad \xi = \frac{L}{L+D}$$

This is a smooth function of η in the vicinity of $\eta = 0$, since $g(r)$ is smooth around $r = 1$, and is maximised at $r = 1$ ($A = L$), as derived in the Yellow Paper and explored using Taylor Series in the Appendix. However, this implies that, even within the Yellow Paper's framework, there would still be a deposit-based arbitrage opportunity if $r < 1$. The drop off to 0 is steep but provided r is sufficiently close to 1, *but still below* 1, this drop is irrelevant. This amounts to having a sufficiently small positive η.

We can manipulate both η and ξ: η using a swap before the deposit and ξ by increasing D. Therefore, there is a small interval of coverage where Platypus's own model states arbitrage is possible but which the penalty is not applied:

$$r \in [r_0(D,L), 1] \quad s.t. \quad \delta(D|r_0,L) = 0$$

This interval *grows* as D increases, as shown in Figure 10 - the larger the deposit size D, the broader the $\delta(D|r_0,L) > 0$ domain and the higher the maximum value $\delta(D|r=1,L)$. Given the definition of $\delta(D|r,L)$ in terms of $g(r)$ and its definition in terms of k, n it follows that $\delta(D|r=1,L) = k\,D$ where Platypus set

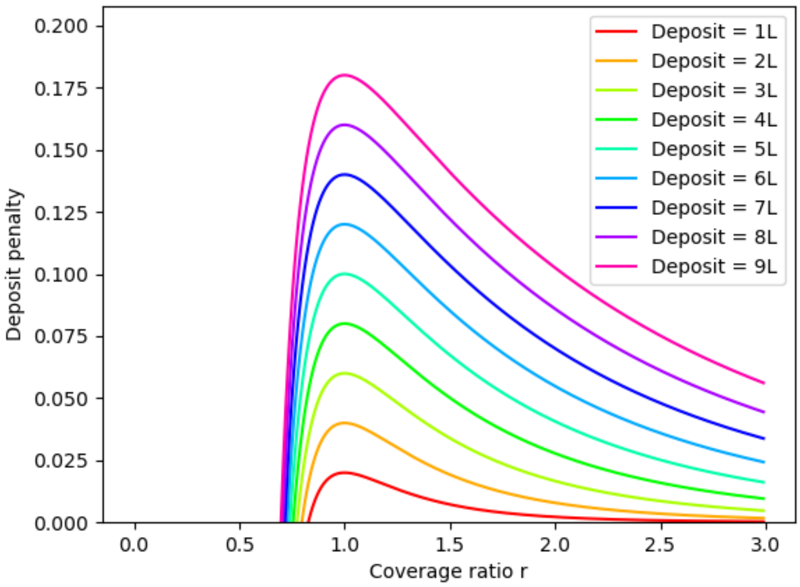

Figure 10 : Deposit penalty function for different deposit sizes. All maximise at $r = 1$ but $r_0(D,L)$ is a monotonically decreasing function of D.

$k = 2 \times 10^{-5}$. Therefore, the maximum grows linearly with the deposit size.

This shows another issue with the Platypus model – it predicts a level of arbitrage that grows with D – D is the liquidity an arbitrager would insert and then remove; any actual arbitrage attack would have its maximum size capped by either the token's assets or liabilities within the pool. This error stems from the incorrect assertion of loop closure and neglecting of D dependent factors within the slippages during the derivation of $\delta(D|r, L)$ in the Yellow Paper.

Even if we ignore this issue and $\delta(D|r, L)$ were an accurate model of potential deposit arbitrage, the use of Indicator Functions is an issue. Due to the Indicator Function it follows that a small change in swap quantity, $\Delta_j^{(1)} \rightarrow \Delta_j^{(1)} + \epsilon$, would lead to a small change in the potential arbitrage value, due to the η-smoothness, but if the swap *just* crosses $r_j = 1$ into $r \in [r_0(D, L), 1)$ the Indicator Function will induce a sudden reduction in penalty, leaving open the potential for arbitrage.

$$\lim_{\epsilon \to 0}[\beta(D|r = 1 + \epsilon, L) - \beta(D|r = 1 - \epsilon, L)] = \delta(D|r = 1, L) = k\ D \gg 0$$

This suggests it may be possible to avoid large penalties or increase slippage benefits through careful calibration of swap quantities to cross the $r = 1$ line via a swap, perform a penalty-free unilateral action, reverse-swap back over $r = 1$ and perform a second penalty-free reverse-unilateral action.

Summary

To summarise our analysis of Platypus's mathematical formalism and arbitrage protection we recap the issues identified:

- Arbitrage loop is not closed in T_j
- Net gain expression misses quadratic term of compounded slippage
- Linear net gain expression's T_j slippage terms incorrectly cancelled
- Nested T_j coverage dependence ignored
- Limited application of deposit/withdrawal penalties

With their arbitrage protection based on an approach containing the above issues it cannot be concluded that Platypus has sufficient protection against arbitrage.

In fact, such an avenue of attack indeed exists.

Arbitrage Attack

Methodology

Consider a StableCoin exchange, such that $f_{i\to j} = f_{j\to i} = 1$ such that pre-arbitrage $r_j > 1 > r_i$. We will perform a Type 2 arbitrage route:

1. $T_i \to T_j$ swap
2. T_j deposit
3. $T_j \to T_i$ reverse swap
4. T_j withdrawal

In this example the major flaw we will exploit is the constrained application of penalties – namely deposit penalties are charged if and only if $r > 1$ and withdrawal penalties if and only if $r < 1$. As a result, *all* the other issues identified need not be exploited here. This serves to give a much simpler illustrative example.

Swap 1

With $r_j^{(0)} > 1$ initially a deposit would incur a penalty so the first swap should reduce r_j below 1. For a given $\epsilon \ll 1$, such as $\epsilon = 10^{-8}$, calculate the $\Delta_i^{(1)}$ such that:

$$r_i^{(1)} = r_i^{(0)} + \frac{\Delta_i^{(1)}}{L_j^{(0)}} \quad \Rightarrow \quad r_j^{(1)} = r_j^{(0)} - \frac{\left(1 - S_{i\to j}^{(1)}\right)\Delta_i^{(1)}}{L_j^{(0)}} = \frac{A_j^{(1)}}{L_j^{(1)}} = 1 - \epsilon < 1$$

The minimal swap size follows:

$$\left(1 - S_{i\to j}^{(1)}\right)\Delta_i^{(1)} > A_j^{(0)} - L_j^{(0)} > 0$$

We have deliberately calibrated the swap to put r_j *just* outside the deposit penalty domain.

Deposit

Deposit a large quantity, D, of T_j, ($D \gg A_j^{(0)}$) which shifts its coverage ratio towards 1 by a tiny amount.

$$r_j^{(2)} = \frac{A_j^{(1)} + D}{L_j^{(1)} + D} = 1 - \epsilon \frac{L_j^{(1)}}{L_j^{(0)} + D} \quad \Rightarrow \quad 1 - \epsilon = r_j^{(1)} < r_j^{(2)} = 1 - \delta < 1$$

No penalty is paid, as intended.

Swap 2

We reverse the swap by selling $\Delta_j^{(2)} T_j$ where $\Delta_j^{(2)}$ is such that r_i returns to its initial value:

$$r_i^{(3)} = r_i^{(1)} - \frac{\left(1 - S_{j\to i}^{(2)}\right)\Delta_j^{(2)}}{L_i^{(0)}} = r_i^{(0)} + \frac{\Delta_i^{(1)}}{L_i^{(0)}} - \frac{\left(1 - S_{j\to i}^{(2)}\right)\Delta_j^{(2)}}{L_i^{(0)}} = r_i^{(0)}$$

This gives the Arbitrage Swap Condition expected: $\Delta_i^{(1)} = \left(1 - S_{j\to i}^{(2)}\right)\Delta_j^{(2)}$. Applying the input side of the swap:

$$r_j^{(3)} = r_j^{(2)} + \frac{\Delta_j^{(2)}}{L_j^{(0)} + D} = \frac{A_j^{(0)} + D - \left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right) - 1\right]\Delta_j^{(2)}}{L_j^{(0)} + D}$$

For larger values of D the change in coverage induced by the swap can be made arbitrarily small:

$$\left|r_j^{(3)} - r_j^{(2)}\right| \ll 1$$

Due to this we can manipulate the slippages over the two swaps to gain net value. Finally, to avoid the withdrawal penalty

To obtain $r_j^{(3)} > 1$ we have a constraint on the assets, liabilities, swap and slippages:

$$A_j^{(0)} - \left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right) - 1\right]\Delta_j^{(2)} > L_j^{(0)}$$

This constraint is dependent on D through $S_{j\to i}^{(2)}$ but so weakly it is not important.

Withdrawal

Finally, we reverse the deposit, withdrawing $D\ T_j$ from the pool. Having ensured $r_j^{(3)} > 1$ we pay no penalty.

Net Gain

The resultant gain is the change in T_j assets:

$$\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right) - 1\right]\Delta_j^{(2)}$$

This is the AVE for zero fees and no penalties, except we achieved it without setting $\alpha = \beta = 0$ but instead avoiding their domains of applicability. Avoiding the withdrawal penalty required $r_j^{(3)} > 1$ and the induced constraint also limits our arbitrage potential:

$$\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right) - 1\right]\Delta_j^{(2)} < A_j^{(0)} - L_j^{(0)} = \left(r_j^{(0)} - 1\right)L_j^{(0)}$$

These two conditions are consistent with one another:

$$(1 - S_{i\to j}^{(1)})\Delta_i^{(1)} = (1 - S_{i\to j}^{(1)})(1 - S_{j\to i}^{(2)})\Delta_j^{(2)} > A_j^{(0)} - L_j^{(0)} > [(1 - S_{i\to j}^{(1)})(1 - S_{j\to i}^{(2)}) - 1]\Delta_j^{(2)}$$

Restructuring:

$$\Delta_j^{(2)} > A_j^{(0)} - L_j^{(0)} - [(1 - S_{i\to j}^{(1)})(1 - S_{j\to i}^{(2)}) - 1]\Delta_j^{(2)} > 0$$

The second swap must be larger than the excess assets, $A_j^{(0)} - L_j^{(0)}$, less the extracted value and the value extracted by this arbitrage route cannot exceed the excess assets.

This shows that if the arbitrate route gives net positive gain it can be repeated so long as the T_j coverage ratio remains above 1.

The implication of this is that we can arbitrage *all* over-covered ($r > 1$) tokens, thanks to the unilateral liquidity of Platypus, down to being exactly covered, $r = 1$. The total yield is then the exchange's entire over-covered liquidity:

$$\text{Total Arbitrage Yield} = \sum_j max(A_j - L_j, 0) = \sum_j max(r_j - 1, 0)L_j$$

Due to the slow rate of arbitrage, as shown in the next section, this would not be practical as the swap volumes would be conspicuously/prohibitively large and frequent but, never-the-less, if *any* token is over-covered it is an arbitrage opportunity.

Numerical Example

Initial exchange configuration:

- T_i : $A_i^{(0)} = 9{,}000$ $L_i^{(0)} = 10{,}000$ $r_i^{(0)} = 0.9$
- T_j : $A_j^{(0)} = 11{,}000$ $L_j^{(0)} = 10{,}000$ $r_j^{(0)} = 1.1$

Swap 1

- $\Delta_i^{(1)} = 999.889218$
- $S_i^{(1)} = -0.00021816$ $S_j^{(1)} = -0.00009736$ so $S_{i\to j}^{(1)} = -0.00012080$
- $\Delta_j^{(1)} = 1{,}000.01$

Therefore

- T_i : $A_i^{(1)} = 9{,}999.889$ $L_i^{(1)} = 10{,}000$ $r_i^{(1)} = 0.9999889$
- T_j : $A_j^{(1)} = 9{,}999.990$ $L_j^{(1)} = 10{,}000$ $r_j^{(1)} = 0.9999990$

Deposit

- $D = 1{,}100{,}000$
- $T_j :$ $A_j^{(2)} = 1{,}109{,}999.990$ $L_j^{(2)} = 1110{,}000$ $r_j^{(2)} = 0.999999990990991$

Swap 2

- $\Delta_j^{(2)} = 999.96788605$
- $S_j^{(2)} = -0.00013950$ $S_i^{(2)} = -0.00021817$ so $S_{j\to i}^{(2)} = 0.00007867$
- $\Delta_i^{(1)} = 999.889$

Therefore the T_i loop is closed and $r_j^{(3)} > 1$:

- $T_i :$ $A_i^{(3)} = 9{,}000$ $L_i^{(3)} = 10{,}000$ $r_i^{(3)} = 0.9$
- $T_j :$ $A_j^{(3)} = 1{,}110{,}999.957886$ $L_j^{(3)} = 1{,}110{,}000$ $r_j^{(3)} = 1.0009009$

Withdrawal

- $W = 1{,}100{,}000$
- $T_j :$ $A_j^{(2)} = 10{,}999.95788605$ $L_j^{(2)} = 10{,}000$ $r_j^{(2)} = 1.09999579$

Net Gain

The net gain in the portfolio is 0.04211395 T_j from swaps of size 1000 T_j, so arbitrage occurs at a rate of 1:25,000.

Figure 11 shows the evolution of the two coverage ratios and the associated cumulative value gain within the portfolio.

Discussion

The evolution of the coverage ratios in Figure 11 demonstrates the Type 2 general behaviour necessary to perform arbitrage against Platypus – use a large deposit to "lock" one token's coverage at $r = 1$ and ensure the swaps move in the right directions.

It should be noted this is *exactly* the scenario outlined in the *Deposit Arbitrage* section of the Yellow Paper except that we have picked the swap size to be sufficiently large that the coverage drops below 1 the step before the deposit, thereby completely avoiding the penalty.

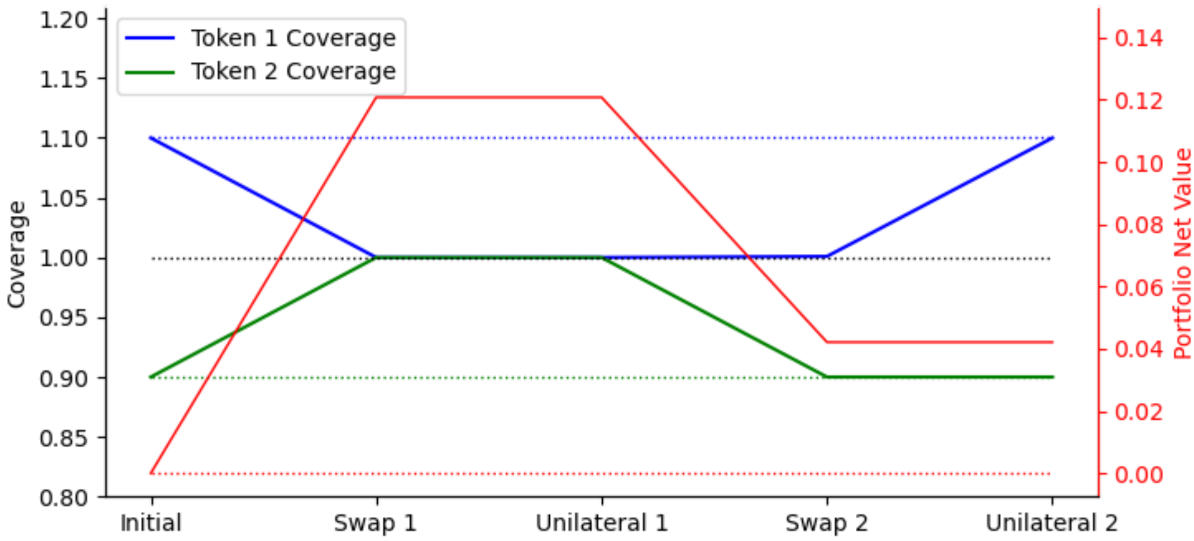

Figure 11 : Coverage evolution during a Type 2 arbitrage route ($T_1 \rightarrow T_2$ swap, T_1 deposit etc) showing the excess net value (Red) after the second (reverse) swap. Total gain 0.04211395 T_1. Initial coverages $r_1 = 1.1$, $r_2 = 0.9$.

This illustrates the error in only applying the deposit penalty for $r > 1$: If we swapped $\Delta_i^{(1)} = 999.869$, instead of $\Delta_i^{(1)} = 999.889218$, the coverage would be too high, $r_i^{(1)} = 1.00001308 > 1$, and the deposit would incur a penalty of $22.000T_1$, completely overwhelming the slippage gained, removing the arbitrage potential. Figure 12 illustrates this - the coverage values are near identical to those of Figure 11 but the tiny change in $r_i^{(1)}$ results in a large penalty.

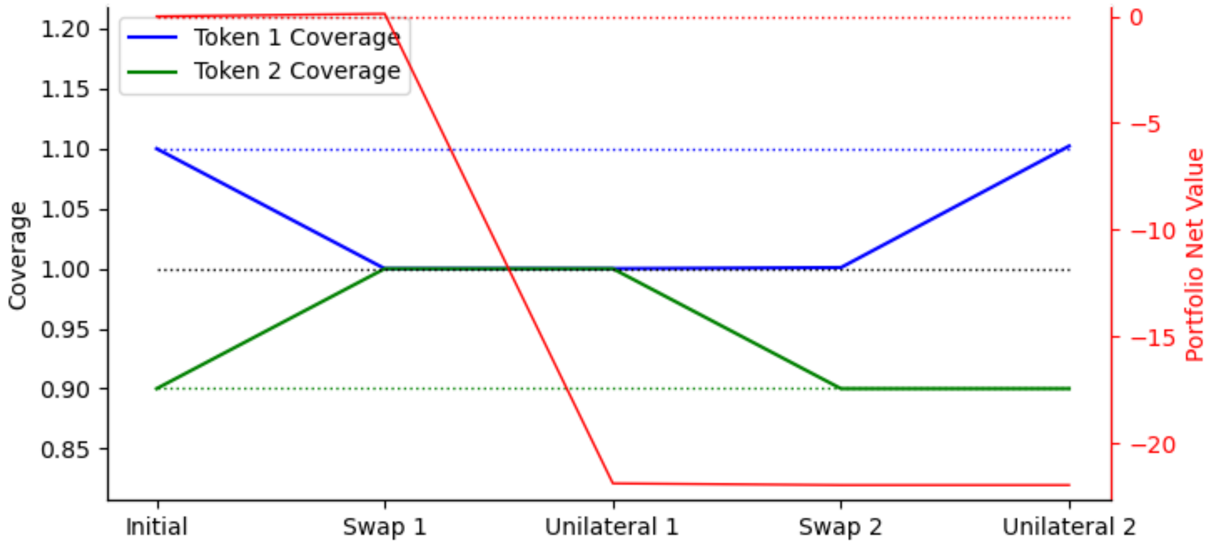

Figure 12 : Arbitrage route almost indistinguishable to Figure 11 coverages except $r_i^{(1)} > 1$, incuring with arbitrage-preventing deposit penalty

Building on Figure 11 we generated Figure 13 using the same Type 2 arbitrage route except the initial coverage of the other token is increased from $r_2 = 0.9$ to $r_2 = 1.0$. The Platypus perspective is blind to this change, for the many reasons previously outlined, but calculating the numerical results shows a small increase in arbitrage returns, 0.0421285 T_1 compared to the previous 0.04211395 T_1. This can only be pushed so far, $r_2 \to \infty$ gives an arbitrage limit of approximately 0.0426121 T_1. Conversely, if $r_2 <$ 0.596 there is a net loss due to the complex interaction of nested and quadratic slippage terms.

For the range $r_2^{(0)} \gg 0.596$ the main effect of varying $r_2^{(0)}$ is to move around where the arbitrage value is gained. Figure 11 shows the first swap gains the value, with the reverse swap removing some of it, while Figure 13 shows the value being gained by the reverse swap.

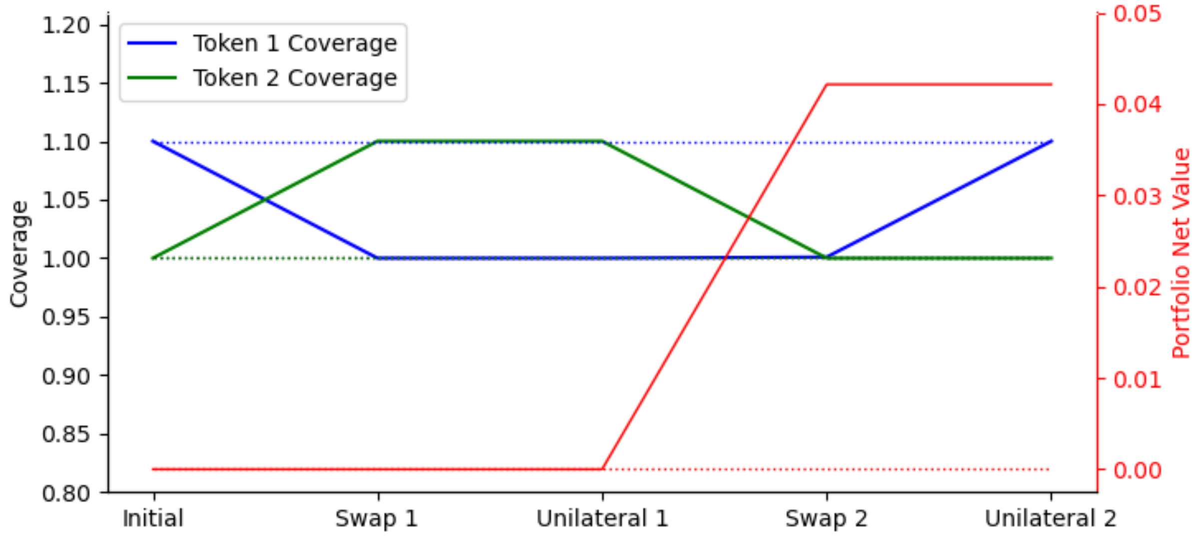

Figure 13 : Coverage evolution during a Type 2 arbitrage route ($T_1 \to T_2$ swap, T_1 deposit etc) showing the excess net value (Red) after the second (reverse) swap. Total gain 0.0421285 T_1. Initial coverages $r_1 = 1.1$, $r_2 = 1.0$

Fee Protection

Fortunately for Platypus this attack vector is *currently* closed but not through the explicit "arbitrage protection" detailed in the Yellow Paper, but instead through fees. In the above numerical process, we set $h = 0$ but Platypus has implemented a 1 basis point fee system, $h = 0.01\%$.

Repeating the above numerical process with fees gives a net gain of -0.157903685 T_1, a loss. A brief piece of trial and error shows the attack vector closes at approximately $h = 0.0021\%$. If Platypus were ever to low their fees to 0.2 basis points, then this attack would be viable, if not a little slow.

Generic Arbitrage

Minimal Fees

$h = 0.0021\%$ is not the minimal level of fees necessary to prevent arbitrage. Small changes can be made by varying T_2 parameters but the primary boost to net arbitrage is found by increasing r_1 for fixed liability L_1.

Repeating the arbitrage attack in the previous section but for $r_1 \gg 1$, such as $L_1^{(0)} = 10{,}000$ and $A_1^{(0)} = 10{,}000{,}000$, then fees of $h = 0.0021\%$ would still allow a total net arbitrage of 924.55369 T_1. The first swap requires 9,990,208 T_2 and the deposit requires 1,000,000,000 T_1 , giving a return rate of approximately 1:10,000 (As high as 1:7,500 if $h = 0$).

To prevent this arbitrage the minimal fees are raised to $h = 0.0067\%$, approximately two-thirds of a basis point. Therefore, if Platypus decided to simply halve its fees, then there would potentially be arbitrage opportunities - though $r \gg 1$ is not a common occurrence now, the nature of the Platypus exchange is that it accumulates "liability free assets" through fees and penalties, causing coverage ratios to slowly drift higher and higher as the exchange matures.

Furthermore, the one sides nature of Platypus's liquidity structure means if *any* token achieves $r \gg 1$ its arbitrage protection will be weaking. This is a feature unique to the unilateral nature of Platypus's liquidity.

Type 4 Arbitrage

Figure 11 and Figure 13 show a generic property of the coverages associated with arbitrage attacks – using the unilateral actions (deposits or withdrawals) to counter some of the "costly slippage". In the Type 2 routes of Figure 11 and Figure 13 we use a large deposit to greatly reduce the movement of r_1 in the reverse swap. Type 2 routes are particularly convenient because it is more natural to do a deposit before a withdrawal *and* we can avoid both deposit and withdrawal penalties entirely while getting value from both swaps.

The same can be made to occur in Type 3 routes:

1. $T_i \rightarrow T_j$ swap such that $r_i^{(0)} < 1$ but $r_i^{(1)} > 1$
2. T_i withdrawal (without penalty) of as much liquidity as possible to cause $r_i^{(2)} \gg r_i^{(1)} > 1$
3. $T_j \rightarrow T_i$ reverse swap such that $r_i^{(3)} < 1$
4. T_i deposit (without penalty)

The coverage and net value gain for a numerical example ($T_2 \rightarrow T_1$ then withdraw T_2) is shown in Figure 14.

- $A_1^{(0)} = 13{,}000$, $L_1^{(0)} = 10{,}000$, $A_2^{(0)} = 9{,}999$, $L_2^{(0)} = 10{,}000$
- Swap 2000 T_2 into 2000.073533 T_1 so that $r_2 \rightarrow 1.19990000$
- Withdraw (without penalty) 9996 T_2 so that $r_2 \rightarrow \frac{2003}{4} = 500.75$
- Swap 2000.073533 T_1 into 2000.14358293 T_2 so that $r_2 \rightarrow 0.7316168$
- Deposit (without penalty) 9996.14358293 T_2

The net gain from this arbitrage route is 0.143583 T_2.

This procedure is less controllable than the Type 2 route – we require $r_i^{(0)} < 1$ but we cannot withdraw more than $A_i^{(0)} < L_i^{(0)}$. The size of the initial swap is also impacted by the availability of the other token, T_j. If the swap depletes T_j too significantly slippages can become costly and prevent arbitrage.

As with Type 2, fees eat into the arbitrage levels. The above numerical example returns a negative amount for $h > 0.0035\%$. Trial and error can easily find situations requiring $h > 0.005\%$ to prevent arbitrage.

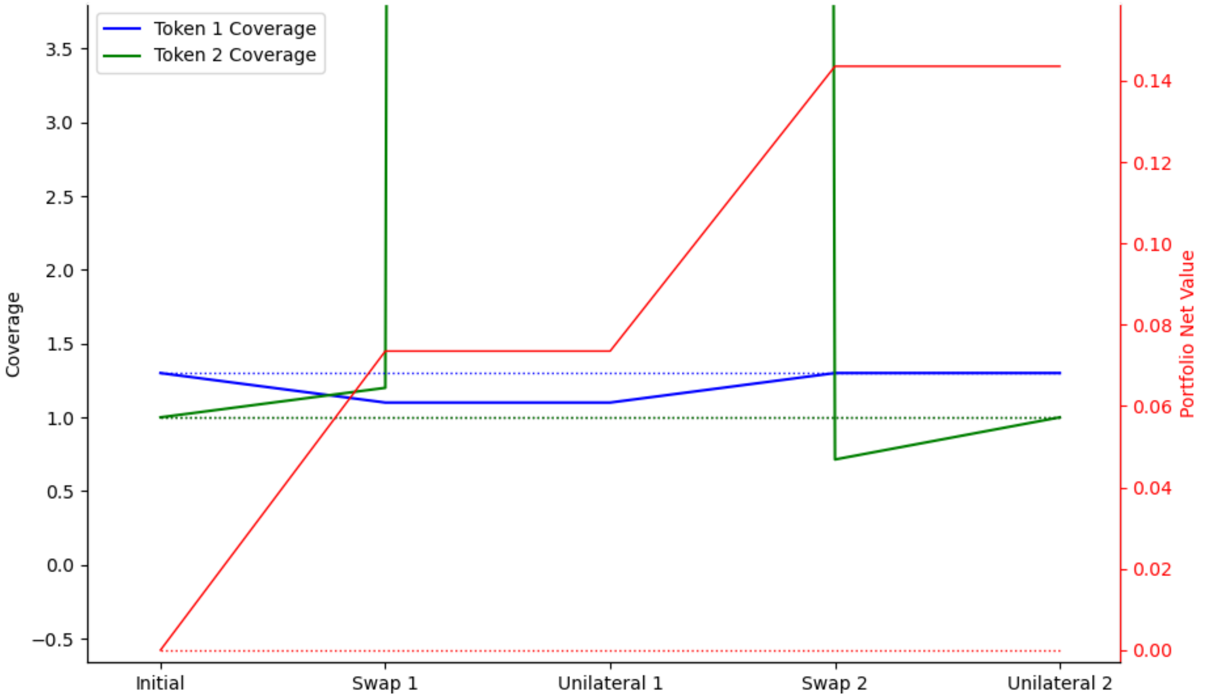

Figure 14 : Type 3 arbitrage route

Type 1 and 4 Arbitrage

Given the general behaviour of Type 2 and Type 3 routes we can infer the behaviour of Type 1 and Type 4 routes. In each case we will be hit by one or even both unilateral penalties. This follows from the fact only swaps can cross the $r = 1$ line. If the initial swap does not cross $r = 1$ then for both unilateral actions either $r > 1$ or $r < 1$, giving one penalty but not both. If the initial swap crosses the $r = 1$ line both penalties will be incurred.

Only in very specific edge cases is arbitrage even potentially possible, namely when the penalty imposed by Platypus does not account for all of the slippage accrued by the arbitrage route. More quantitatively we might want to explore routes where the neglected quadratic term $V_2 = \Delta_j^{(1)}\left(S_{j\to i}^{(1)} S_{i\to j}^{(2)}\right)$ and non-closed marginal $V_{YP,i}$ become relevant. However, the fact Platypus's derivation of its arbitrage protection has a mixture of PSC and ASC consistent expressions, slippage neglection and incorrect parametrisations means we cannot focus on a single term, we must consider the full AVE with α, β replaced by the Platypus penalty expressions.

Since this amounts to a full investigation of the slippage terms we instead transition to a full analysis of, potentially, how to remedy the Platypus arbitrage protection.

Repairing Arbitrage Protection

With all the issues of the Platypus Yellow Paper's derivation of the deposit and withdrawal penalties we cannot correct the problem by just adding a correction term, we should use the full AVE and design the α, β functions to prevent that function ever being positive.

$$V = \Delta_m^{(n)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$$

This is a non-trivial problem for several reasons:

1. Multiple arbitrage routes give different emphasis to deposits versus withdrawals
2. The dependency on *both* tokens' liquidities makes algebraic methods difficult
3. The nested dependency of $S_{j\to i}^{(2)}$ depending on $S_{i\to j}^{(1)}$ also makes algebraic methods difficult
4. The balance between platform convenience and platform stability

Elaborating on Reason 4 – the use of Indicator Functions can be seen to be motivated by the desire to not penalise deposits or withdrawals that increase coverage, as this improves the "health" of the relevant token. We have demonstrated that the penalty function $\delta(D|r, L)$ is non-zero for $r < 1$ but deposits for $r < 1$ increases coverage – would it present a negative image of Platypus if it penalised people improving the exchange's liquidity? But without penalties for $r < 1$ deposits and/or $r > 1$ withdrawals arbitrage cannot be prevented for all h fee values – would it present a negative image of Platypus if its arbitrage protection is purely fee-based, not its stated arbitrage protection?

Using fees is a "blunt tool", since it will also penalise actions that aren't usable for arbitrage, rather than the more targeted approach of deposit/withdrawal specific penalties. With that in mind we look to determine α, β functions that prevent arbitrage even if $h = 0$.

Similarly, as we will do frequently in the coming analysis, there are choices in how to bound slippage values – tighter bounds would make for a cheaper platform but are more mathematically elaborate. We will give mathematically convenient bounds, whether these align with the aims of Platypus is a secondary matter.

Ultimately, we will not provide a tight, closed form bound for arbitrage protection penalty functions but rather outline the procedure required and the mathematical challenges that arise from the non-triviality reasons listed above – a highly efficient working arbitrage protection is beyond the scope of this article.

Slippage Behaviour

We first derive the generic behaviour for slippage, $S_{i \to j} = S_i - S_j$:

$$S_i = S(r_i, r_i + dr_i) = \frac{g(r_i + dr_i) - g(r_i)}{dr_i}$$

With $g(r)$ monotonically decreasing, this function is always negative. The gradient itself, $g'(r)$, is monotonically non-decreasing, with minimum value $g'(r) = 1$ for $r \le r^*$. Since dr can be negative, as well as positive, we write $r_i^- = min(r_i, r_i + dr_i)$ and $r_i^+ = max(r_i, r_i + dr_i)$, though $S(r, r') = S(r', r)$ makes the order irrelevant.

Therefore, since S_i is the mean gradient over the interval $[r_i^-, r_i^+]$ it is bounded by the gradient on the interval boundaries:

$$S_i = \frac{g(r_i + dr_i) - g(r_i)}{dr_i} \in [g'(r_i^-), g'(r_i^+)]$$

A second useful bound follows from $g(r)$ properties:

$$S_i \ge -\frac{g(r_i^-)}{r_i^+ - r_i^-}$$

Combining these inequalities gives us a variety of tighter or looser bounds:

$$g'(r_i^-) \le -\frac{g(r_i^-)}{r_i^+ - r_i^-} \le S_i = S(r_i^-, r_i^+) \le g'(r_i^+) \le 0$$

The geometric origins of these inequalities are shown in Figure 15 for a generic monotonic decreasing weakly convex positive function.

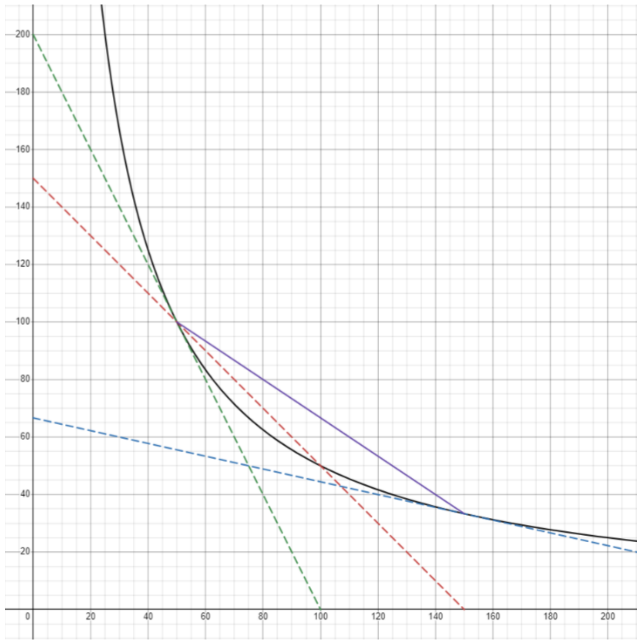

Figure 15 : Marginal slippage bounds. $g'(r_i^-)$ *(Green),* $-\frac{g(r_i^-)}{r_i^+-r_i^-}$ *(Red),* S_i *(Purple),* $g'(r_i^+)$ *(Blue). Example function used* $y=\frac{1}{x}$.

Using these bounds for the unilateral slippage term we can bound the full slippage:

$$\min(S_i - S_j) = min(S_i) - max(S_j) \quad , \quad \max(S_i - S_j) = max(S_i) - min(S_j)$$

This offers several options about how to combine multi-variate bounds, as we can tighten or loosen the different bounds separately. Generically we have:

$$min(S_i) - max(S_j) \ \le S_i - S_j \le max(S_i) - min(S_j)$$

Highlighting some of the more mathematically convenient versions:

$$g'(r_i^-) \ \le g'(r_i^-) \ - \ g'(r_j^+) \le -\frac{g(r_i^-)}{r_i^+ - r_i^-} - \ g'(r_j^+) \le S_i - S_j$$

$$S_i - S_j \ \le g'(r_i^+) + \frac{g(r_j^-)}{r_j^+ - r_j^-} \le g'(r_i^+) - g'(r_j^-) \le \ -g'(r_j^-)$$

Therefore, we can generate a loose set of bounds for slippage that each depend on only one variable:

$$-1 \ \le g'(r_i^-) \le S_{i\to j} \le -g'(r_j^-) \ \le +1$$

This is a weakened version of a bivariate set of bounds:

$$-1 \ \le g'(r_i^-) \ - \ g'(r_j^+) \le S_{i\to j} \le g'(r_i^+) - g'(r_j^-) \ \le +1$$

Arbitrage Value Bounding

Using these bounds, we can approximate the worst-case values for the arbitrage value. Since the only relevant part is the slippage terms, we focus on that

$$V_{\text{Slip}} = \left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right) - 1 = -S_{i\to j}^{(1)} - S_{j\to i}^{(2)} + S_{i\to j}^{(1)} S_{j\to i}^{(2)}$$

Since both $S_{i\to j}^{(1)}, S_{j\to i}^{(2)} < 0$ for gain we can find a worst-case upper bound by replacing the slippages by their "most negative bound" that has both coverage variables.

We now insert the coverage values dependent on the swaps and unilateral actions. For $S_{i\to j}^{(1)}$ this is straightforward but the form of $S_{j\to i}^{(2)}$ depends on the arbitrage route we use. Considering $S_{i\to j}^{(1)}$ first we use the coverage formulae derived earlier, and the Appendix:

- $S_{i\to j}^{(1)}$
    - $r_i^- = \frac{A_i^{(0)}}{L_i^{(0)}}$
    - $r_i^+ = \frac{A_i^{(0)}}{L_i^{(0)}} + \frac{\Delta_i^{(1)}}{L_i^{(0)}}$
    - $r_j^- = \frac{A_j^{(0)}}{L_j^{(0)}} - f_{i\to j}\frac{\Delta_i^{(1)}}{L_j^{(0)}}$
    - $r_j^+ = \frac{A_j^{(0)}}{L_j^{(0)}}$

Inserting into the slippage:

$$-S_{i\to j}^{(1)} \leq \ +g'\left(r_j^+\right) - g'\left(r_i^-\right) = +\,g'\left(\frac{A_j^{(0)}}{L_j^{(0)}}\right) - g'\left(\frac{A_i^{(0)}}{L_i^{(0)}}\right)$$

This expression has now dependence on the swap size, its only dependency would be as an overall multiplicative factor scaling V_{Slip} in the AVE. For completeness, a tighter alternative bound is

$$-S_{i\to j}^{(1)} \leq +\,g'\left(r_j^+\right) + \frac{g\left(r_i^-\right)}{r_i^+ - r_i^-} = +\,g'\left(\frac{A_j^{(0)}}{L_j^{(0)}}\right) + \frac{L_j^{(0)}}{f_{i\to j}\Delta_i^{(1)}}\, g\left(\frac{A_j^{(0)}}{L_j^{(0)}}\right)$$

Since $S_{j\to i}^{(2)}$ reverses this swap, by increasing T_j assets and decreasing T_i assets, we know which expressions are maximum or minimum:

- $S_{j\to i}^{(2)}$
    - $r_i^- = \frac{A_i^{(2)}}{L_i^{(2)}} - \frac{f_{j\to i}\Delta_1^{(2)}}{L_i^{(2)}}$

- $r_i^+ = \frac{A_i^{(2)}}{L_i^{(2)}}$
- $r_j^- = \frac{A_j^{(2)}}{L_j^{(2)}}$
- $r_j^+ = \frac{A_j^{(2)}}{L_j^{(2)}} + \frac{\Delta_j^{(2)}}{L_j^{(2)}}$

The four arbitrage routes will give rise to different expressions for post-unilateral action $A_i^{(2)}$, $L_i^{(2)}$, $A_j^{(2)}$, $L_j^{(2)}$ configurations. Nevertheless, we can bound the arbitrage values in terms of these intermediate configurations:

$$-S_{j\to i}^{(2)} \leq +g'(r_i^+) - g'(r_j^-) = +g'\left(\frac{A_i^{(2)}}{L_i^{(2)}}\right) - g'\left(\frac{A_j^{(2)}}{L_j^{(2)}}\right)$$

The tighter, swap volume dependent, version is

$$-S_{j\to i}^{(2)} \leq +g'(r_i^+) + \frac{g(r_j^-)}{r_j^+ - r_j^-} = +g'\left(\frac{A_i^{(2)}}{L_i^{(2)}}\right) + \frac{L_j^{(2)}}{f_{j\to i}\Delta_j^{(2)}} g\left(\frac{A_j^{(2)}}{L_j^{(2)}}\right)$$

The choice of arbitrage route manifests in which term is dependent on *either* D or W but we can still consider the coverage dependency of these bounds.

$$dr \geq 0 \quad \Rightarrow \quad g'(r) \leq g'(r + dr) \quad , \quad g(r) \leq g(r - dr)$$

Therefore, for all $dr_{i,j} > 0$ we can obtain weaker bounds for $-S_{j\to i}^{(2)}$ as follows:

$$-S_{j\to i}^{(2)} \leq g'\left(\frac{A_i^{(2)}}{L_i^{(2)}}\right) - g'\left(\frac{A_j^{(2)}}{L_j^{(2)}}\right) \leq g'\left(\frac{A_i^{(2)}}{L_i^{(2)}} + dr_i\right) - g'\left(\frac{A_j^{(2)}}{L_j^{(2)}} - dr_j\right)$$

$$-S_{j\to i}^{(2)} \leq g'\left(\frac{A_i^{(2)}}{L_i^{(2)}}\right) + \frac{L_j^{(2)}}{f_{j\to i}\Delta_j^{(2)}} g\left(\frac{A_j^{(2)}}{L_j^{(2)}}\right) \leq g'\left(\frac{A_i^{(2)}}{L_i^{(2)}} + dr_i\right) + \frac{L_j^{(2)}}{f_{j\to i}\Delta_j^{(2)}} g\left(\frac{A_j^{(2)}}{L_j^{(2)}} - dr_j\right)$$

Using the simplest bounds for $-S_{j\to i}^{(2)}$ we can now bound V_{Slip} as a whole:

$$V_{\text{Slip}} \leq +g'\left(\frac{A_j^{(0)}}{L_j^{(0)}}\right) - g'\left(\frac{A_i^{(0)}}{L_i^{(0)}}\right) + g'\left(\frac{A_i^{(2)}}{L_i^{(2)}}\right) - g'\left(\frac{A_j^{(2)}}{L_j^{(2)}}\right) + \left[g'\left(\frac{A_j^{(0)}}{L_j^{(0)}}\right) - g'\left(\frac{A_i^{(0)}}{L_i^{(0)}}\right)\right]\left[g'\left(\frac{A_i^{(2)}}{L_i^{(2)}}\right) - g'\left(\frac{A_j^{(2)}}{L_j^{(2)}}\right)\right]$$

As we shall shortly see, these bounds are algebraically simple but extremely inefficient, their purpose is purely to illustrate a general method of arbitrage protection design.

Worst Case Penalties

In the Appendix (Alternative Arbitrage Routes – General Route) we derive a general arbitrage route, including the associated fees, where the step between the two swaps is composed of *two* unilateral actions of general sign, U_i, U_j, one for each token, so we can simultaneously consider all four arbitrage routes.

- $\frac{A_i^{(2)}}{L_i^{(2)}} = \frac{A_i^{(0)}+\Delta_i^{(1)}+U_i+\alpha(U_i)I_-(U_i)}{L_i^{(0)}+U_i-\beta(U_i)I_+(U_i)} = \frac{A_i^{(1)}+U_i+\alpha(U_i)I_-(U_i)}{L_i^{(1)}+U_i-\beta(U_i)I_+(U_i)}$
- $\frac{A_j^{(2)}}{L_j^{(2)}} = \frac{A_j^{(0)}-f_{i\to j}\left(1-S_{i\to j}^{(1)}\right)\Delta_i^{(1)}+U_j+\alpha(U_j)I_-(U_j)}{L_j^{(0)}+U_j-\beta(U_j)I_+(U_j)} = \frac{A_j^{(1)}+U_j+\alpha(U_j)I_-(U_j)}{L_j^{(1)}+U_j-\beta(U_j)I_+(U_j)}$

We can also make the AVE's α, β dependency manifest:

- T_j loop closure enforced in T_i units:

$$V_{|i} = \Delta_i^{(1)} V_{\text{Slip}} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)\Omega\left(U_j, V_j\right) - \Omega(U_i, V_i)$$

- T_i loop closure enforced in T_j units:

$$V_{|j} = \Delta_j^{(2)} V_{\text{Slip}} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)\Omega(U_i, V_i) - \Omega\left(U_j, V_j\right)$$

This immediately highlights a problem that was neglected in the Platypus Yellow Paper – α, β self reference by appearing in Ω $and\ S_{j\to i}^{(2)}$. The specific forms of α, β are required to keep the extracted value V always negative but the value depends on α, β.

In the Appendix also we demonstrate the most viable way to perform arbitrage is to use one of the 4 arbitrage route types we have defined, simultaneous deposits or withdrawals in *both* tokens only generates more costs. With arbitrage slippage boosting the associated penalty the 4 routes zero out the slippage boosted Ω term.

Applying this arbitrage restriction to the AVE expressions and the coverage expressions:

- T_j loop closure enforced in T_i units (Type 1 and Type 3):

$$V_{|i} = \Delta_i^{(1)} V_{\text{Slip}} - \Omega(U_i, V_i) \quad , \quad r_j^{(2)} = \frac{A_j^{(2)}}{L_j^{(2)}} = \frac{A_j^{(0)}}{L_j^{(0)}} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)\frac{\Delta_i^{(1)}}{L_j^{(0)}}$$

- T_i loop closure enforced in T_j units (Type 2 and Type 4):

$$V_{|j} = \Delta_j^{(2)} V_{\text{Slip}} - \Omega\left(U_j, V_j\right) \quad , \quad r_i^{(2)} = \frac{A_i^{(2)}}{L_i^{(2)}} = \frac{A_i^{(0)}}{L_i^{(0)}} + \frac{\Delta_i^{(1)}}{L_i^{(0)}}$$

It is clear the arbitrage value $V_{|i,j}$ is expressible entirely in terms of the initial configurations, $A_{i,j}^{(0)}$, $L_{i,j}^{(0)}$, the swap size $\Delta_{i,j}^{(1,2)}$ and the unilateral action $U_{i,j}$:

$$V_{|i,j} = V_{|i,j}\left(A_i^{(0)}, A_j^{(0)}, L_i^{(0)}, L_j^{(0)}, \Delta, U\right)$$

However, as stated in the Appendix, the penalties can only depend on the state of the pool at the instant of the unilateral action and the size/direction of the unilateral action itself, $A_{i,j}^{(1)}$, $L_{i,j}^{(1)}$, $U_{i,j}$. This is seen in the Platypus Yellow Paper, where the arbitrage penalty is set by finding the initial configurations and swap volume that result in the maximum arbitrage value and using that value as the penalty. The full case is made more difficult by our full accounting for slippage, the self-referencing penalty and dependency on *both* token configurations but the premise is the same.

Overall, the penalty for the first unilateral action should be equal to the extractable value of the worst arbitrage route that has the $A_{i,j}^{(1)}$, $L_{i,j}^{(1)}$ configuration at the instant of applying the unilateral action $U_{i,j}$.

Quantitative Example

For example, in a Type 1 arbitrage route we have the following AVE:

$$V_{|i} = \Delta_i^{(1)} V_{\text{Slip}}\left(A_i^{(0)}, A_j^{(0)}, L_i^{(0)}, L_j^{(0)}, \Delta_i^{(1)}, U_i\right) - \beta_1\left(U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}\right) - \alpha_1\left(V_i A_i^{(3)}, A_j^{(3)}, L_i^{(3)}, L_j^{(3)}\right)$$

It is sufficient to design the Type 1 β, β_1, to counter the positive arbitrage, we will design α using a different arbitrage route so we restrict our attention to the case of $\alpha_1 = 0$ in this expression.

To align arguments, we note $L_{i,j}^{(0)} = L_{i,j}^{(1)}$, leaving $A_{i,j}^{(0,1)}$ and $\Delta_i^{(1)}$ to align. For the T_i token the relationship is simple, $A_i^{(0)} = A_i^{(1)} - \Delta_i^{(1)}$. The T_j case is complicated by the non-linear slippage effect:

$$A_j^{(1)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\left(A_i^{(0)}, A_j^{(0)}, L_i^{(0)}, L_j^{(0)}, \Delta_i^{(1)}\right)\right)\Delta_i^{(1)}$$

Inverting this relationship into $A_j^{(0)} = \mathrm{F}\left(A_j^{(1)} \middle| A_i^{(1)}, L_i^{(0)}, L_j^{(0)}, \Delta_i^{(1)}\right)$ is possible but not in a closed form expression. Putting this issue aside for a moment we define the deposit arbitrage protection as follows:

$$\beta_1\left(U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}\right) \equiv \max_{\Delta_i^{(1)}}\left[\Delta_i^{(1)}\, V_{\text{Slip}}\left(U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}, \Delta_i^{(1)}\right)\right]$$

$\Delta_i^{(1)}$ is parameterising a family of Type 1 arbitrage routes that pass through the specific $A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}$ configuration and have a first unilateral action of a U_i deposit of T_i tokens.

From the 6 parameters $U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}, \Delta_i^{(1)}$ we can calculate the assets and liabilities $A_{i,j}^{(0,1,2,3,4)}$, $L_{i,j}^{(0,1,2,3,4)}$ at all steps of the Type 1 route, replace V_{Slip} with the g' based bounds derived above and maximise over $\Delta_i^{(1)}$.

$$\frac{A_i^{(2)}}{L_i^{(2)}} = \frac{A_i^{(1)} + U_i}{L_i^{(1)} + U_i - \beta(U_i)} \quad , \quad \frac{A_j^{(2)}}{L_j^{(2)}} = \frac{A_j^{(1)}}{L_j^{(1)}}$$

Inserting the calculated configurations into the slippage bounds:

$$-S_{i\to j}^{(1)} \leq g'\left(\frac{F\left(A_j^{(1)} \middle| A_i^{(1)}, L_i^{(1)}, L_j^{(1)}, \Delta_i^{(1)}\right)}{L_j^{(1)}}\right) - g'\left(\frac{A_i^{(1)} - \Delta_i^{(1)}}{L_i^{(1)}}\right)$$

$$-S_{j\to i}^{(2)} \leq g'\left(\frac{A_i^{(1)} + U_i}{L_i^{(1)} + U_i - \beta_1\left(U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}\right)}\right) - g'\left(\frac{A_j^{(1)}}{L_j^{(1)}}\right)$$

We again arrive at a self-reference issue, since β_1 impacts the reverse swap's slippage, which is part of its own definition.

Due to the nonlinearity of slippages, self-referential definitions, and quadratic slippage arbitrage expressions it is inevitable either very loose bounds or numerical optimisation is required. Since we only wish to outline the procedure, we address the algebraic inconveniences of both slippage terms by weakening their bounds further by noting $g'(r) < 0$ for all r.

$$-S_{i\to j}^{(1)} \leq -g'\left(\frac{A_i^{(1)} - \Delta_i^{(1)}}{L_i^{(1)}}\right) \quad , \quad -S_{j\to i}^{(2)} \leq -g'\left(\frac{A_j^{(1)}}{L_j^{(1)}}\right)$$

This weakening completely removes the dependency on the T_i deposit volume U_i. Putting these into the AVE slippage expression:

$$V_{\text{Slip}} \leq -g'\left(\frac{A_i^{(1)} - \Delta_i^{(1)}}{L_i^{(1)}}\right) - g'\left(\frac{A_j^{(1)}}{L_j^{(1)}}\right) + g'\left(\frac{A_i^{(1)} - \Delta_i^{(1)}}{L_i^{(1)}}\right) g'\left(\frac{A_j^{(1)}}{L_j^{(1)}}\right)$$

Immediate from this expression we have bounded the Type 1 swap volume, $\Delta_i^{(1)} \leq A_i^{(1)}$ – intuitively obvious as the pool cannot have just received more T_i tokens than it currently holds. Recalling the β definition:

$$\beta_1\left(U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}\right) \equiv \max_{\Delta_i^{(1)}}\left[\Delta_i^{(1)} V_{\text{Slip}}\left(U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}, \Delta_i^{(1)}\right)\right]$$

The $\Delta_i^{(1)}$ dependency in the V_{Slip} bound makes maximising this expression simple as the weak bounds we have used make the T_j factors swap independent. The T_i dependent $-g'(r)$ is maximised for $r \leq r^*$,

$-g'(r \leq r^*) = +1$, and we push the swap to be as large as possible to maximise the overall $\Delta_i^{(1)}$ coefficient, giving $\Delta_i^{(1)} = A_i^{(1)}$.

The interpretation of this is a swap that sells a volume of $A_i^{(1)}$ T_i tokens to a pool with *zero* T_i tokens. This aligns with intuition since the Platypus slippage is designed to significantly reward such an action and it cannot have less than 0 assets in a token.

$$\beta_1\left(U_i, A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}\right) \equiv A_i^{(1)}\left[+1 - 2\, g'\left(\frac{A_j^{(1)}}{L_j^{(1)}}\right)\right]$$

This is an *extreme* penalty, equivalent to replacing $-S_{i\to j}^{(1)} = +1$ earlier in the analysis, which certainly avoids the issues with nonlinear slippage and self-referential penalties.

Generically the arbitrage analysis outlined here would be repeated for all 4 arbitrage route types but without taking very weak approximations/bounds[11]. The two deposit-based routes will each give an expression for β, while the withdrawal-based routes will each give an expression for α. Though the parametrisation of the family of routes via the swap volume will be different for each, due to which token experiences slippage, the above analysis suggests they will otherwise give the same equations in terms of the pre-unilateral configurations $A_i^{(1)}, A_j^{(1)}, L_i^{(1)}, L_j^{(1)}$.

[11] Such an analysis would be prohibitively lengthy, even for this article!

Summary

In this article we have explored the Platypus exchange platform in detail, identifying an arbitrage attack vector through mathematical analysis, confirmed its existence via numerical examples and finally explored the avenues might be taken by the Platypus development team to close this vulnerability.

Platypus approaches the challenge of Automated Market Makers in quite a unique fashion and the general mathematical foundation allows for a rich variation of price slippage models and address significant issues in other AMMs, such as liquidity fragmentation.

However, while the general “marginal slippage function” framework allows many results to be proven in general, the analysis required to create an *efficient* penalty model to protect against arbitrage is considerable.

The nonlinear dependencies of the slippage, self-referential nature of penalty definitions and compounded slippages force the need for either approximations or considerable use of numerical methods when designing arbitrage protection. Approximations may allow for costs to be clearer; it will likely make costs higher. Numerical methods can make pricing models more efficient, increasing the utility of a pool, but also make it harder to confirm full arbitrage protection has been achieved.

It will be interesting to see how the Platypus development team will strike a balance between these competing factors as the platform continues to evolve but, as a priority, the lack of full general arbitrage protection should be dealt with first.

Appendix

Alternative Arbitrage Routes

We define the 4 following inequivalent "potential arbitrage routes" for 2 tokens:

- Type 1: swap $T_i \to T_j$, deposit T_i, swap $T_j \to T_i$, withdraw T_i
- Type 2: swap $T_i \to T_j$, deposit T_j, swap $T_j \to T_i$, withdraw T_j
- Type 3: swap $T_i \to T_j$, withdraw T_i, swap $T_j \to T_i$, deposit T_i
- Type 4: swap $T_i \to T_j$, withdraw T_j, swap $T_j \to T_i$, deposit T_j

In the main body of this article, we have done a step-by-step walkthrough of Type 1, where the "swapped in" token is then also deposited. We will do one further step-by-step walkthrough on the "opposite" case, Type 4, which withdraws the "swapped out" token. For the remaining two we give exchange state evolution without detailed derivation.

Type 4 Route

- Swap: T_i into T_j: $\left(r_i^{(0)}, r_j^{(0)}\right) \to \left(r_i^{(1)}, r_j^{(1)}\right)$ where $r_i^{(1)} > r_i^{(0)}$ and $r_j^{(1)} < r_j^{(0)}$
- Unilateral: Withdraw a quantity W of T_j, $r_j^{(1)} \to r_j^{(2)}$ (with $r_i^{(2)} = r_i^{(1)}$)
- Swap T_2 into T_i: $\left(r_i^{(2)}, r_j^{(2)}\right) \to \left(r_i^{(3)}, r_j^{(3)}\right)$ where $r_i^{(3)} < r_i^{(2)}$ and $r_j^{(3)} > r_j^{(2)}$
- Reverse unilateral: Deposit a quantity D of T_j, $r_j^{(3)} \to r_j^{(4)}$ (with $r_i^{(4)} = r_i^{(3)}$)

Since we only care about the net change we set $X_i^{(0)} = X_j^{(0)} = Y_i^{(0)} = Y_j^{(0)} = 0$.

Initial Swap

The trader sells $\Delta_i^{(1)} T_i$ to the pool. We calculate slippage using to determine the terminal exchange rate.

- Pool In:
  - Amount: $\Delta_i^{(1)}$ of T_i
  - Assets: $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - Liability: $L_i^{(1)} = L_i^{(0)}$
  - Cost-less coverage: $r_i^{(1)} = \frac{A_i^{(1)}}{L_i^{(1)}} = r_i^{(0)} + \frac{\Delta_i^{(1)}}{L_i^{(0)}}$
  - Marginal slippage: $S_i^{(1)} = \frac{g\left(r_i^{(1)}\right) - g\left(r_i^{(0)}\right)}{r_i^{(1)} - r_i^{(0)}} = \frac{L_i^{(0)}}{\Delta_i^{(1)}}\left(g\left(r_i^{(1)}\right) - g\left(r_i^{(0)}\right)\right)$
- Pool Out:
  - Amount: $f_{i \to j} \Delta_i^{(1)}$ of T_j
  - Assets: $A_j^{(1)} = A_j^{(0)} - f_{i \to j} \Delta_i^{(1)}$

- Liability: $L_j^{(1)} = L_j^{(0)}$
- Cost-less coverage: $r_j^{(1)} = \frac{A_j^{(1)}}{L_j^{(1)}} = r_j^{(0)} - f_{i \to j} \frac{\Delta_i^{(1)}}{L_j^{(0)}}$
- Marginal slippage: $S_j^{(1)} = \frac{g\left(r_j^{(1)}\right) - g\left(r_j^{(0)}\right)}{r_j^{(1)} - r_j^{(0)}} = -\frac{L_i^{(0)}}{f_{i \to j}\,\Delta_i^{(1)}}\left(g\left(r_j^{(1)}\right) - g\left(r_j^{(0)}\right)\right)$

Generate full slippage and exchange rates:

- Slippage $S_{i \to j}^{(1)} = S_i^{(1)} - S_j^{(1)} = \frac{L_i^{(0)}}{\Delta_i^{(1)}}\left(g\left(r_i^{(1)}\right) - g\left(r_i^{(0)}\right)\right) + \frac{L_i^{(0)}}{f_{i \to j}\,\Delta_i^{(1)}}\left(g\left(r_j^{(1)}\right) - g\left(r_j^{(0)}\right)\right)$
- Terminal Exchange Rate $f_{i \to j}^{*} = f_{i \to j}\left(1 - S_{i \to j}^{(1)}\right)(1 - h)$

The actual swap using the terminal exchange rate is then performed:

- Pool In:
  - Amount: $\Delta_i^{(1)}$ of T_i
  - Assets: $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - Liability: $L_i^{(1)} = L_i^{(0)}$
  - Coverage: $r_i^{(1)} = \frac{A_i^{(1)}}{L_i^{(1)}} = r_i^{(0)} + \frac{\Delta_i^{(1)}}{L_i^{(0)}}$
- Pool Out:
  - Amount: $f_{i \to j}\left(1 - S_{i \to j}^{(1)}\right)(1 - h)\Delta_i^{(1)}$ of T_j
  - Assets: $A_j^{(1)} = A_j^{(0)} - f_{i \to j}\left(1 - S_{i \to j}^{(1)}\right)(1 - h)\Delta_i^{(1)}$
  - Liability: $L_j^{(1)} = L_j^{(0)}$
  - Coverage: $r_j^{(1)} = \frac{A_j^{(1)}}{L_j^{(1)}} = r_j^{(0)} - \frac{f_{i \to j}\left(1 - S_{i \to j}^{(1)}\right)(1 - h)\Delta_i^{(1)}}{L_j^{(0)}}$

The trader's assets change in the opposite manner:

- Trader out = Pool in $X_i^{(1)} = X_i^{(0)} - dA_i^{(0)} = -\Delta_i^{(1)}$
- Trader in = Pool out $X_j^{(1)} = X_j^{(0)} - dA_j^{(0)} = +f_{i \to j}\left(1 - S_{i \to j}^{(1)}\right)(1 - h)\Delta_i^{(1)}$

Withdrawal

The trader redeems a set of Platypus tokens from their portfolio, totalling $\mathrm{W}C_j$, for T_j, potentially paying a penalty α if $r_j^{(1)} < 1$.

- Pool Out:
  - Amount: $\mathrm{W} - \alpha$ of T_j
  - Assets: $A_j^{(2)} = A_j^{(1)} - (\mathrm{W} - \alpha)$

- Liability: $L_j^{(2)} = L_j^{(1)} - \mathrm{W}$
- Coverage: $r_j^{(2)} = \frac{A_j^{(1)} - (\mathrm{W} - \alpha)}{L_j^{(1)} - \mathrm{W}}$

All other pool quantities are unchanged. The trader's portfolio will change in the opposite manner:

- Trader out = Pool in: $Y_j^{(2)} = Y_j^{(1)} - \mathrm{W}$
- Trader in = Pool out: $X_j^{(2)} = X_j^{(1)} + (\mathrm{W} - \alpha)$

All other quantities for the trader and pool are unchanged.

Reverse Swap

A second swap, in the opposite direction, is then performed by the trader, selling $\Delta_j^{(2)}$ T_j to the pool.

- Pool In:
  - Amount: $\Delta_j^{(2)}$ of T_j
  - Assets: $A_j^{(2)} \to A_j^{(3)} = A_j^{(2)} + \Delta_j^{(2)}$
  - Liability: $L_j^{(2)} \to L_j^{(3)} = L_j^{(2)}$
  - Coverage: $r_j^{(2)} \to r_j^{(3)} = \frac{A_j^{(3)}}{L_j^{(3)}} = \frac{A_j^{(2)} + \Delta_j^{(2)}}{L_j^{(2)}} = r_j^{(2)} + \frac{\Delta_j^{(2)}}{L_j^{(2)}}$
  - Marginal slippage: $S_j^{(2)} = \frac{g\left(r_j^{(3)}\right) - g\left(r_j^{(2)}\right)}{r_j^{(3)} - r_j^{(2)}} = \frac{L_j^{(2)}}{\Delta_j^{(2)}}\left(g\left(r_j^{(3)}\right) - g\left(r_j^{(2)}\right)\right)$
- Pool Out:
  - Amount: $f_{j\to i}\Delta_j^{(2)}$ of T_i
  - Assets: $A_i^{(2)} \to A_i^{(3)} = A_i^{(2)} - f_{j\to i}\Delta_j^{(2)}$
  - Liability: $L_i^{(2)} \to L_i^{(3)} = L_i^{(2)}$
  - Coverage: $r_i^{(2)} \to r_i^{(3)} = \frac{A_i^{(3)}}{L_i^{(3)}} = \frac{A_i^{(2)} - f_{j\to i}\Delta_j^{(2)}}{L_i^{(2)}} = r_i^{(2)} - \frac{f_{j\to i}\Delta_j^{(2)}}{L_i^{(2)}}$
  - Marginal slippage: $S_i^{(1)} = \frac{g\left(r_i^{(1)}\right) - g\left(r_i^{(0)}\right)}{r_i^{(1)} - r_i^{(0)}} = -\frac{L_i^{(2)}}{f_{j\to i}\,\Delta_j^{(2)}}\left(g\left(r_i^{(3)}\right) - g\left(r_i^{(2)}\right)\right)$

Generate full slippage and exchange rates:

- Slippage $S_{j\to i}^{(2)} = S_j^{(2)} - S_i^{(2)} = \frac{L_j^{(2)}}{\Delta_j^{(2)}}\left(g\left(r_j^{(3)}\right) - g\left(r_j^{(2)}\right)\right) + \frac{L_i^{(2)}}{f_{j\to i}\,\Delta_j^{(2)}}\left(g\left(r_i^{(3)}\right) - g\left(r_i^{(2)}\right)\right)$
- Terminal Exchange Rate $f_{j\to i}^{*} = f_{j\to i}\left(1 - S_{j\to i}^{(1)}\right)(1 - h)$

The actual swap using the terminal exchange rate is then performed:

- Pool In:
  - Amount: $\Delta_j^{(2)}$ of T_j
  - Assets: $A_j^{(2)} \to A_j^{(3)} = A_j^{(2)} + \Delta_j^{(2)}$
  - Liability: $L_j^{(2)} \to L_j^{(3)} = L_j^{(2)}$
  - Coverage: $r_j^{(2)} \to r_j^{(3)} = \frac{A_j^{(3)}}{L_j^{(3)}} = \frac{A_j^{(2)}+\Delta_j^{(2)}}{L_j^{(2)}} = r_j^{(2)} + \frac{\Delta_j^{(2)}}{L_j^{(2)}}$
- Pool Out:
  - Amount: $f_{j\to i}\left(1 - S_{j\to i}^{(1)}\right)(1-h)\,\Delta_j^{(2)}$ of T_i
  - Assets: $A_i^{(2)} \to A_i^{(3)} = A_i^{(2)} - f_{j\to i}\left(1 - S_{j\to i}^{(1)}\right)(1-h)\,\Delta_j^{(2)}$
  - Liability: $L_i^{(2)} \to L_i^{(3)} = L_i^{(2)}$
  - Coverage: $r_i^{(2)} \to r_i^{(3)} = \frac{A_i^{(3)}}{L_i^{(3)}} = r_i^{(2)} - \frac{f_{j\to i}\left(1-S_{j\to i}^{(1)}\right)(1-h)\,\Delta_j^{(2)}}{L_1^{(2)}}$

The trader's assets change in the opposite manner:

- Trader out = Pool in $X_j^{(3)} = X_j^{(2)} - dA_j^{(2)} = X_j^{(2)} - \Delta_j^{(2)}$
- Trader in = Pool out $X_i^{(3)} = X_i^{(2)} - dA_i^{(2)} = X_i^{(2)} + f_{j\to i}\left(1 - S_{j\to i}^{(1)}\right)(1-h)\,\Delta_j^{(2)}$

Deposit

The trader deposits $D\ T_j$ to the pool, which could experience a penalty of β if $r_j^{(1)} > 1$.

- Pool In:
  - Amount: D of T_j
  - Assets: $A_j^{(4)} = A_j^{(3)} + D$
  - Liability: $L_j^{(4)} = L_j^{(3)} + (D - \beta)$
  - Coverage: $r_j^{(4)} = \frac{A_j^{(3)}+D}{L_j^{(3)}+(D-\beta)}$

To "compensate" the trader the pool will mint a set of smart tokens corresponding to the change in liability.

- Pool Out:
  - Amount: $D - \beta$ of C_j

The trader's assets change in the opposite manner:

- Trader out = Pool in: $X_j^{(2)} = X_j^{(1)} - D$

- Trader in = Pool out: $Y_j^{(2)} = Y_j^{(1)} + (D - \beta)$

In order to be part of an arbitrage sequence D must result in the newly acquired C_j tokens *exactly* cancelling out the deficit generated by the withdrawal:

$$0 = Y_j^{(3)} + (D - \beta) \quad \Rightarrow \quad D = W + \beta$$

This requires accounting for the penalty β, which is straight forward to do. All other quantities for the trader and pool are unchanged.

Evolution

Using the above results, we summarise the evolution of the pool assets and liabilities in terms of its initial state and the 4 actions.

- Pool T_i values
  - $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - $A_i^{(2)} = A_i^{(0)} + \Delta_i^{(1)}$
  - $A_i^{(3)} = A_i^{(0)} + \Delta_i^{(1)} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$
  - $A_i^{(4)} = A_i^{(0)} + \Delta_i^{(1)} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$
  - $L_i^{(1)} = L_i^{(0)}$
  - $L_i^{(2)} = L_i^{(0)}$
  - $L_i^{(3)} = L_i^{(0)}$
  - $L_i^{(4)} = L_i^{(0)}$
- Pool T_j values
  - $A_j^{(1)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$
  - $A_j^{(2)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} - (\mathrm{W} - \alpha)$
  - $A_j^{(3)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} - (\mathrm{W} - \alpha) + \Delta_j^{(2)}$
  - $A_j^{(4)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} - (\mathrm{W} - \alpha) + \Delta_j^{(2)} + D$
  - $L_j^{(1)} = L_j^{(0)}$
  - $L_j^{(2)} = L_j^{(0)} - \mathrm{W}$
  - $L_j^{(3)} = L_j^{(0)} - \mathrm{W}$
  - $L_j^{(4)} = L_j^{(0)}$

The portfolio's end state has zero $C_{i,j}$ positions so we need only consider its $T_{i,j}$ holdings.

- Portfolio T_i values
  - $X_i^{(1)} = -\Delta_i^{(1)}$

- $X_i^{(2)} = -\Delta_i^{(1)}$
- $X_i^{(3)} = -\Delta_i^{(1)} + f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$
- $X_i^{(4)} = -\Delta_i^{(1)} + f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$

- Portfolio T_2 values
  - $X_j^{(1)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$
  - $X_j^{(2)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} + (\mathrm{W} - \alpha)$
  - $X_j^{(3)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} + (\mathrm{W} - \alpha) - \Delta_j^{(2)}$
  - $X_j^{(4)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} + (\mathrm{W} - \alpha) - \Delta_j^{(2)} - (W + \beta)$

The total net change, in T_2 units, of the final portfolio status:

$$V = f_{i\to j}\Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)(1-h) - 1\right] + \Delta_j^{(2)}\left[\left(1 - S_{j\to i}^{(2)}\right)(1-h) - 1\right] - (\alpha + \beta)$$

Swap Constraint for Arbitrage

Closing the T_i sequence into a loop for arbitrage:

$$X_i^{(4)} = 0 \quad \Rightarrow \quad \Delta_i^{(1)} = f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_2^{(2)}$$

Applying this to the portfolio's arbitrage value:

$$V = \Delta_j^{(2)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$$

Type 2 Route

We consider the following arbitrage route:

- Swap: T_i into T_j: $\left(r_i^{(0)}, r_j^{(0)}\right) \to \left(r_i^{(1)}, r_j^{(1)}\right)$ where $r_i^{(1)} > r_i^{(0)}$ and $r_j^{(1)} < r_j^{(0)}$
- Unilateral: Deposit a quantity D of T_j, $r_j^{(1)} \to r_j^{(2)}$ (with $r_i^{(2)} = r_i^{(1)}$)
- Swap T_j into T_i: $\left(r_i^{(2)}, r_j^{(2)}\right) \to \left(r_i^{(3)}, r_j^{(3)}\right)$ where $r_i^{(3)} < r_i^{(2)}$ and $r_j^{(3)} > r_j^{(2)}$
- Reverse unilateral: Deposit a quantity W of T_j, $r_j^{(3)} \to r_j^{(4)}$ (with $r_i^{(4)} = r_i^{(3)}$)

We summarise the evolution of the pool assets and liabilities:

- Pool T_i values
  - $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - $A_i^{(2)} = A_i^{(0)} + \Delta_i^{(1)}$
  - $A_i^{(3)} = A_i^{(0)} + \Delta_i^{(1)} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$
  - $A_i^{(4)} = A_i^{(0)} + \Delta_i^{(1)} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$
  - $L_i^{(1)} = L_i^{(0)}$

- $L_i^{(2)} = L_i^{(0)}$
- $L_i^{(3)} = L_i^{(0)}$
- $L_i^{(4)} = L_i^{(0)}$

- Pool T_j values
  - $A_j^{(1)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)}$
  - $A_j^{(2)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} + D$
  - $A_j^{(3)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} + D + \Delta_j^{(2)}$
  - $A_j^{(4)} = A_j^{(0)} - f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} + D + \Delta_j^{(2)} - \big((D-\beta) - \alpha\big)$
  - $L_j^{(1)} = L_j^{(0)}$
  - $L_j^{(2)} = L_j^{(0)} + (D - \beta)$
  - $L_j^{(3)} = L_j^{(0)} + (D - \beta)$
  - $L_j^{(4)} = L_j^{(0)}$

The portfolio's end state has zero $C_{i,j}$ positions so we need only consider its $T_{i,j}$ holdings.

- Portfolio T_i values
  - $X_i^{(1)} = -\Delta_i^{(1)}$
  - $X_i^{(2)} = -\Delta_i^{(1)}$
  - $X_i^{(3)} = -\Delta_i^{(1)} + f_{j\to i}\big(1 - S_{j\to i}^{(2)}\big)(1-h)\,\Delta_j^{(2)}$
  - $X_i^{(4)} = -\Delta_i^{(1)} + f_{j\to i}\big(1 - S_{j\to i}^{(2)}\big)(1-h)\,\Delta_j^{(2)}$
- Portfolio T_j values
  - $X_j^{(1)} = +f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)}$
  - $X_j^{(2)} = +f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} - D$
  - $X_j^{(3)} = +f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} - D - \Delta_j^{(2)}$
  - $X_j^{(4)} = +f_{i\to j}\big(1 - S_{i\to j}^{(1)}\big)(1-h)\,\Delta_i^{(1)} - D - \Delta_j^{(2)} + \big((D-\beta) - \alpha\big)$

The portfolio net value change, in units of T_j:

$$V = \Delta_j^{(2)}\big[\big(1 - S_{j\to i}^{(2)}\big)(1-h) - 1\big] + f_{i\to j}\Delta_i^{(1)}\big[\big(1 - S_{i\to j}^{(1)}\big)(1-h) - 1\big] - (\alpha + \beta)$$

The loop closing condition:

$$\Delta_i^{(1)} = f_{j\to i}\big(1 - S_{j\to i}^{(2)}\big)(1-h)\,\Delta_j^{(2)}$$

The portfolio's arbitrage net value change, in units of T_j:

$$V = \Delta_j^{(2)}\big[\big(1 - S_{i\to j}^{(1)}\big)\big(1 - S_{j\to i}^{(2)}\big)(1-h)^2 - 1\big] - (\alpha + \beta)$$

Type 3 Route

We consider the following arbitrage route:

- Swap: T_i into T_j: $\left(r_i^{(0)}, r_j^{(0)}\right) \rightarrow \left(r_i^{(1)}, r_j^{(1)}\right)$ where $r_j^{(1)} < r_j^{(0)}$ and $r_i^{(1)} > r_i^{(0)}$
- Unilateral: Withdraw a quantity W of T_i, $r_i^{(1)} \rightarrow r_i^{(2)}$ (with $r_j^{(2)} = r_j^{(1)}$)
- Swap T_2 into T_i: $\left(r_i^{(2)}, r_j^{(2)}\right) \rightarrow \left(r_i^{(3)}, r_j^{(3)}\right)$ where $r_i^{(3)} < r_i^{(2)}$ and $r_j^{(3)} > r_j^{(2)}$
- Reverse unilateral: Deposit a quantity D of T_i, $r_i^{(3)} \rightarrow r_i^{(4)}$ (with $r_j^{(4)} = r_j^{(3)}$)

We summarise the evolution of the pool assets and liabilities:

- Pool T_i values
    - $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
    - $A_i^{(2)} = A_i^{(0)} + \Delta_i^{(1)} - (\mathrm{W} - \alpha)$
    - $A_i^{(3)} = A_i^{(0)} + \Delta_i^{(1)} - (\mathrm{W} - \alpha) - f_{j\rightarrow i}\left(1 - S_{j\rightarrow i}^{(2)}\right)(1 - h)\ \Delta_j^{(2)}$
    - $A_i^{(4)} = A_i^{(0)} + \Delta_i^{(1)} - (\mathrm{W} - \alpha) - f_{j\rightarrow i}\left(1 - S_{j\rightarrow i}^{(2)}\right)(1 - h)\ \Delta_j^{(2)} + (W + \beta)$
    - $L_i^{(1)} = L_i^{(0)}$
    - $L_i^{(2)} = L_i^{(0)} - \mathrm{W}$
    - $L_i^{(3)} = L_i^{(0)} - \mathrm{W}$
    - $L_i^{(4)} = L_i^{(0)}$
- Pool T_j values
    - $A_j^{(1)} = A_j^{(0)} - f_{i\rightarrow j}\left(1 - S_{i\rightarrow j}^{(1)}\right)(1 - h)\Delta_i^{(1)}$
    - $A_j^{(2)} = A_j^{(0)} - f_{i\rightarrow j}\left(1 - S_{i\rightarrow j}^{(1)}\right)(1 - h)\Delta_i^{(1)}$
    - $A_j^{(3)} = A_j^{(0)} - f_{i\rightarrow j}\left(1 - S_{i\rightarrow j}^{(1)}\right)(1 - h)\Delta_i^{(1)} + \Delta_j^{(2)}$
    - $A_j^{(4)} = A_j^{(0)} - f_{i\rightarrow j}\left(1 - S_{i\rightarrow j}^{(1)}\right)(1 - h)\Delta_i^{(1)} + \Delta_j^{(2)}$
    - $L_j^{(1)} = L_j^{(0)}$
    - $L_j^{(2)} = L_j^{(0)}$
    - $L_j^{(3)} = L_j^{(0)}$
    - $L_j^{(4)} = L_j^{(0)}$

The portfolio's end state has zero $C_{i,j}$ positions so we need only consider its $T_{i,j}$ holdings.

- Portfolio T_i values
    - $X_i^{(1)} = -\Delta_i^{(1)}$
    - $X_i^{(2)} = -\Delta_i^{(1)} + (\mathrm{W} - \alpha)$
    - $X_i^{(3)} = -\Delta_i^{(1)} + (\mathrm{W} - \alpha) + f_{j\rightarrow i}\left(1 - S_{j\rightarrow i}^{(2)}\right)(1 - h)\ \Delta_j^{(2)}$

- $X_i^{(4)} = -\Delta_i^{(1)} + (\mathrm{W} - \alpha) + f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)} - (W + \beta)$

- Portfolio T_j values
  - $X_j^{(1)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$
  - $X_j^{(2)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$
  - $X_j^{(3)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} - \Delta_j^{(2)}$
  - $X_j^{(4)} = +f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} - \Delta_j^{(2)}$

The total net change, in T_i units, of the final portfolio status:

$$V = \Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)(1-h) - 1\right] + f_{j\to i}\Delta_j^{(2)}\left[\left(1 - S_{j\to i}^{(2)}\right)(1-h) - 1\right] - (\alpha + \beta)$$

Closing the T_j sequence into a loop for arbitrage:

$$X_j^{(4)} = 0 \quad \Rightarrow \quad \Delta_j^{(2)} = f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$$

Applying this to the portfolio's arbitrage value:

$$V = \Delta_i^{(1)}\left[\left(1 - S_{j\to i}^{(2)}\right)\left(1 - S_{i\to j}^{(1)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$$

Routes summary

Collating the results and abstracting to general token labels.

Type 1: T_i into T_j then deposit T_i

- $V = \Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$
- $\Delta_j^{(2)} = f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$

Type 2: T_i into T_j then deposit T_j

- $V = \Delta_j^{(2)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$
- $\Delta_i^{(1)} = f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$

Type 3: T_i into T_j then withdraw T_i

- $V = \Delta_i^{(1)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$
- $\Delta_j^{(2)} = f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$

Type 4: T_i into T_j then withdraw T_j

- $V = \Delta_j^{(2)}\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$
- $\Delta_i^{(1)} = f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_2^{(2)}$

The general expression only depends on the token that is acted upon unilaterally, not the unilateral action itself. Instead that will impact the evolution of the coverages and the value of the penalties.

General Route

In full generality for an arbitrage route starting T_i into T_j then unilateral on T_k we have *Arbitrage Value Equation (AVE)*:

$$V = \Delta\left[\left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2 - 1\right] - (\alpha + \beta)$$

This suppresses the dependencies of the swap sizes, slippages and penalties. To make these manifest we consider a *general route* where the unilateral actions are now done to *both* tokens by a generic amount. For example, T_i is modified by a unilateral change U_i where $U_i < 0$ is a withdrawal and $U_i > 0$ is a deposit. We then account for the associated penalties using indicator functions:

$$I_-(x) = 1 \quad \text{iff} \quad x < 0 \quad , \quad I_+(x) = 1 \quad \text{iff} \quad x > 0$$

The route is therefore

- Swap: T_i into T_j: $\left(r_i^{(0)}, r_j^{(0)}\right) \to \left(r_i^{(1)}, r_j^{(1)}\right)$ where $r_j^{(1)} < r_j^{(0)}$ and $r_i^{(1)} > r_i^{(0)}$
- Double Unilateral:
  - Unilateral change U_i T_i
  - Unilateral change U_j T_j
- Swap T_j into T_i: $\left(r_i^{(2)}, r_j^{(2)}\right) \to \left(r_i^{(3)}, r_j^{(3)}\right)$ where $r_i^{(3)} < r_i^{(2)}$ and $r_j^{(3)} > r_j^{(2)}$
- Double Unilateral:
  - Unilateral change V_i T_i
  - Unilateral change V_j T_j

We summarise the evolution of the pool assets and liabilities:

- Pool T_i values
  - $A_i^{(1)} = A_i^{(0)} + \Delta_i^{(1)}$
  - $A_i^{(2)} = A_i^{(1)} + U_i + \alpha(U_i) I_-(U_i)$
  - $A_i^{(3)} = A_i^{(2)} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$
  - $A_i^{(4)} = A_i^{(3)} + V_i + \alpha(\mathrm{V_i})\, I_-(V_i)$
  - $L_i^{(1)} = L_i^{(0)}$
  - $L_i^{(2)} = L_i^{(1)} + U_i - \beta(U_i) I_+(U_i)$
  - $L_i^{(3)} = L_i^{(2)}$
  - $L_i^{(4)} = L_i^{(3)} + V_i - \beta(V_i) I_+(V_i)$
- Pool T_j values

- $A_j^{(1)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$
- $A_j^{(2)} = A_j^{(1)} + U_j + \alpha(U_j)I_-(U_j)$
- $A_j^{(3)} = A_j^{(2)} + \Delta_j^{(2)}$
- $A_j^{(4)} = A_j^{(3)} + V_j + \alpha(V_j)I_-(V_j)$
- $L_j^{(1)} = L_j^{(0)}$
- $L_j^{(2)} = L_j^{(1)} + U_j - \beta(U_j)I_+(U_j)$
- $L_j^{(3)} = L_j^{(2)}$
- $L_j^{(4)} = L_j^{(3)} + V_j - \beta(V_j)I_+(V_j)$

Combining these to get the final state of each parameter:

- $A_i^{(4)} = A_i^{(0)} + \Delta_i^{(1)} + U_i + \alpha(U_i)I_-(U_i) - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)} + V_i + \alpha(\mathrm{V_i})\,I_-(V_i)$
- $L_i^{(4)} = L_i^{(0)} + U_i - \beta(U_i)I_+(U_i) + V_i - \beta(V_i)I_+(V_i)$
- $A_j^{(4)} = A_j^{(0)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} + U_j + \alpha(U_j)I_-(U_j) + \Delta_j^{(2)} + V_j + \alpha(V_j)I_-(V_j)$
- $L_j^{(4)} = L_j^{(0)} + U_j - \beta(U_j)I_+(U_j) + V_j - \beta(V_j)I_+(V_j)$

Closing the liability loops requires constraining the unilateral actions to one another:

- $L_i^{(4)} - L_i^{(0)} = 0 \quad \Rightarrow \quad U_i + V_i = \beta(U_i)I_+(U_i) + \beta(V_i)I_+(V_i)$
- $L_j^{(4)} - L_j^{(0)} = 0 \quad \Rightarrow \quad U_j + V_j = \beta(U_j)I_+(U_j) + \beta(V_j)I_+(V_j)$

Each of the 4 arbitrage routes corresponds to a specific $I_\pm(U_{i,j})$ indicator function being non-zero.

For convenient we define the following penalty function:

$$\Omega(U,V) = \alpha(U)I_-(U) + \alpha(V)I_-(V) + \beta(U)I_+(U) + \beta(V)I_+(V)$$

In this combination, under the constraint $U + V = \beta(U)I_+(U) + \beta(V)I_+(V)$, and for generic *non-zero* U, V exactly two of the terms, one an α, the other a β, will be non-zero. This also corresponds to the generic penalty in the AVE.

Using these to simplify the asset changes:

- $A_i^{(4)} - A_i^{(0)} = +\Delta_i^{(1)} - f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)} + \Omega(U_i, V_i)$
- $A_j^{(4)} - A_j^{(0)} = +\Delta_j^{(2)} - f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} + \Omega(U_j, V_j)$

Arbitrage routes Type 1 and Type 3 set $U_j = V_j = \Omega(U_j, V_j) = 0$ as the unilateral actions are applied to T_i and so set $A_j^{(4)} - A_j^{(0)} = 0$ to give their ASC:

$$\Delta_j^{(2)} = f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)}$$

Correspondingly arbitrage routes Type 2 and Type 4 set $U_i = V_i = \Omega(U_i, V_i) = 0$ as the unilateral actions are applied to T_j and set $A_i^{(4)} - A_i^{(0)} = 0$ to give their ASC:

$$\Delta_i^{(1)} = f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)}$$

Inserting these ASCs into the other asset change gives the corresponding (negative) AVE. We can generalise this now by including the $\Omega(U, V)$ term necessary to close the relevant token loop.

Double unilateral Type 2 and Type 4 T_i loop closure:

$$A_i^{(4)} - A_i^{(0)} = 0 \quad \Rightarrow \quad \Delta_i^{(1)} = f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\,\Delta_j^{(2)} - \Omega(U_i, V_i)$$

The negative AVE then becomes

$$\Delta_j^{(2)}\left(1 - \left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2\right) + f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Omega(U_i, V_i) + \Omega\left(U_j, V_j\right)$$

Double unilateral Type 1 and Type 3 T_j loop closure:

$$A_j^{(4)} - A_j^{(0)} = 0 \quad \Rightarrow \quad \Delta_j^{(2)} = f_{i\to j}\left(1 - S_{i\to j}^{(1)}\right)(1-h)\Delta_i^{(1)} - \Omega\left(U_j, V_j\right)$$

The negative AVE then becomes

$$\Delta_i^{(1)}\left(1 - \left(1 - S_{i\to j}^{(1)}\right)\left(1 - S_{j\to i}^{(2)}\right)(1-h)^2\right) + f_{j\to i}\left(1 - S_{j\to i}^{(2)}\right)(1-h)\Omega\left(U_j, V_j\right) + \Omega(U_i, V_i)$$

Though this "generalises" the loop closure it can be excluded for several reasons. In both versions of the negative AVE the Ω coefficients are positive and since $\Omega(U, V) \geq 0$ this means they always hinder arbitrage potential. Furthermore, the slippage necessary for arbitrage also increases the effects of one of the penalty terms. Therefore, the best arbitrate route is to *not* perform a double unilateral action, restricting to one of the four "single unilateral action" sequences, specifically the one not multiplied by a slippage factor.

With this in mind, we specify the Ω expressions for each of the arbitrage routes:

- Type 1:
    - $U_i > 0$, $V_i < 0$, $U_j = 0$, $V_j = 0$
    - $\Omega(U_i, V_i) = \beta(U_i) + \alpha(V_i)$
    - $\Omega\left(U_j, V_j\right) = 0$
- Type 2:
    - $U_i = 0$, $V_i = 0$, $U_j > 0$, $V_j < 0$
    - $\Omega(U_i, V_i) = 0$
    - $\Omega\left(U_j, V_j\right) = \beta\left(U_j\right) + \alpha\left(V_j\right)$
- Type 3:
    - $U_i < 0$, $V_i > 0$, $U_j = 0$, $V_j = 0$

  - $\Omega(U_i, V_i) = \alpha(U_i) + \beta(V_i)$
  - $\Omega(U_j, V_j) = 0$
- Type 4:
  - $U_i = 0$, $V_i = 0$, $U_j < 0$, $V_j > 0$
  - $\Omega(U_i, V_i) = 0$
  - $\Omega(U_j, V_j) = \alpha(U_j) + \beta(V_j)$

The AVE with manifest dependencies then takes two forms, depending on which token has the unilateral action:

- T_j loop closure AVE in T_i units due to unilateral T_i action, effectively rendering $\Omega(U_j, V_j) = 0$

$$V_{|i} = \Delta_i^{(1)} \left((1 - S_{i\to j}^{(1)})(1 - S_{j\to i}^{(2)})(1-h)^2 - 1 \right) - \Omega(U_i, V_i)$$

- T_i loop closure AVE in T_j units due to unilateral T_j action, effectively rendering $\Omega(U_i, V_i) = 0$

$$V_{|j} = \Delta_j^{(2)} \left((1 - S_{i\to j}^{(1)})(1 - S_{j\to i}^{(2)})(1-h)^2 - 1 \right) - \Omega(U_j, V_j)$$

We can elaborate on the dependency of α, β by noting the state of the pool when the unilateral action is applied, and the loop closure constrains the swap amounts:

- $\beta(U_i) \to \beta(U_i, A_i^{(1)}, L_i^{(1)}, A_j^{(1)}, L_j^{(1)})$
- $\beta(U_j) \to \beta(U_j, A_i^{(1)}, L_i^{(1)}, A_j^{(1)}, L_j^{(1)})$
- $\alpha(U_i) \to \alpha(U_i, A_i^{(1)}, L_i^{(1)}, A_j^{(1)}, L_j^{(1)})$
- $\alpha(U_j) \to \alpha(U_j, A_i^{(1)}, L_i^{(1)}, A_j^{(1)}, L_j^{(1)})$
- $\beta(V_i) \to \beta(V_i, A_i^{(3)}, L_i^{(3)}, A_j^{(3)}, L_j^{(3)})$
- $\beta(V_j) \to \beta(V_j, A_i^{(3)}, L_i^{(3)}, A_j^{(3)}, L_j^{(3)})$
- $\alpha(V_i) \to \alpha(V_i, A_i^{(3)}, L_i^{(3)}, A_j^{(3)}, L_j^{(3)})$
- $\alpha(V_j) \to \alpha(V_j, A_i^{(3)}, L_i^{(3)}, A_j^{(3)}, L_j^{(3)})$

These follow from the fact $U_{i,j}$ is the first unilateral action after the first swap and $V_{i,j}$ occurs after the reverse swap. The penalties cannot know the state of the pool at any time other than the moment of the unilateral action, therefore they cannot explicitly depend on swap amounts $\Delta_{i,j}^{(1,2)}$ or the initial configurations, $A_{i,j}^{(0)}$, $L_{i,j}^{(0)}$.

Platypus Deposit Penalty

The general function given in the Yellow Paper is

$$\delta(D|r,L) = g\left(\frac{rL+D}{L+D}\right)(L+D) - g(r)L$$

Given a pre-deposit coverage of $r = 1 - \eta$ the post-deposit coverage is

$$r \;\rightarrow \frac{rL+D}{L+D} = 1 + \frac{(r-1)L}{L+D} = 1 - \frac{\eta\, L}{L+D} \equiv 1 - \eta\xi \quad \Rightarrow \quad \xi = \frac{L}{L+D}$$

The penalty function is non-zero in some range $r \in [r_-, r_+]$ such that $1 \in [r_-, r_+]$. To estimate $r_\pm$ we perform a Taylor expansion:

$$\delta(D|r,L) = \left[g(1) - \xi\eta\, g'(1) + \frac{1}{2}(-\xi\eta)^2\right](L+D) - \left[g(1) - \eta\, g'(1) + \frac{1}{2}(-\eta)^2\right]L$$

Collecting powers of η:

- $\eta^0 : g(1)\, D$
- $\eta^1 : g'(1)\eta[-\xi(L+D)+L] = g'(1)\eta\left[-(L+D)\frac{L}{L+D} + L\right] = 0$
- $\eta^1 : \frac{1}{2}g''(1)\eta^2[\xi^2(L+D) - L] = \frac{1}{2}g''(1)\eta^2\left[(L+D)\left(\frac{L}{L+D}\right)^2 - L\right] = -\frac{1}{2}g''(1)\eta^2\frac{DL}{L+D}$

Collecting the results:

$$\delta(D|r = 1+\eta, L) = g(1)D - \frac{1}{2}g''(1)\eta^2\frac{DL}{L+D} + O(\eta^3)$$

This shows $\eta = 0$ is a local maximum, with value read off from the η^0 term or just from the original definition of $\delta(D|r,L)$:

$$\delta(D|r = 1, L) = g(1)D = kD = 0.00002D$$