HydraDX **Omnipool** Specification and Economic Report

BlockScience

March 22, 2022

Contents

Disclaimer i

Preface ii

Notation . . . ii

Formatting . . . iii

1 Executive Summary 1

2 Introduction 1

3 The Omnipool Specification 3

3.1 State Space Representation . . . 3

3.1.1 Global state (System Level) . . . 3

3.1.2 Local State (Agent Level) . . . 4

3.2 System Constraints . . . 5

3.2.1 Total pool shares . . . 5

3.3 Mechanisms . . . 5

3.3.1 Add Asset Liquidity (single asset, existing token) . . . 5

3.3.2 Add Asset Liquidity (single asset, new token) . . . 8

3.3.3 Withdraw Asset Liquidity (single asset) . . . 10

3.3.4 Swapping LRNA with asset i . . . 16
3.3.5 Swap asset i with asset j (sell side) . . . 20
3.3.6 Swap asset i with asset j (buy side) . . . 23

4 Economic Implications **26**
4.1 Pool Viability . . . 26
4.1.1 Exponential Withdrawal . . . 26
4.1.2 Finite Price Convergence . . . 26
4.1.3 Reciprocal Swaps . . . 28
4.1.4 Discretionary Pool Shares . . . 29
4.1.5 HDX Support . . . 29
4.2 Impermanent Loss . . . 32
4.2.1 IL without transactions fees . . . 32
4.2.2 IL mitigation from fees . . . 35
4.3 Slippage . . . 37
4.3.1 Slippage without fees . . . 38
4.3.2 Slippage with fees . . . 39

Disclaimer

This work is based upon the **Omnipool** specification and mechanisms conveyed to BlockScience by HydraDX between January and March 2022. It does not *in any way* reflect the code implementation of the **Omnipool**, which has not been communicated to BlockScience and for which BlockScience is not responsible. The specification contained herein is descriptive and does not constitute the reference specification document for the **Omnipool**. Its assumptions regarding the specification form and associated mechanisms may not be relied upon to reflect the final specification form, associated mechanisms or implementation of the **Omnipool**.

This work does not constitute financial advice of any kind, including investment advice. It does not advocate for or against any particular automated market maker framework, and by referencing cryptographically-supported tokens or fiat currencies does not act in any way as a recommendation for or against investment in said tokens or fiat currencies.

Conditional upon its assumptions every effort has been made to ensure that the analysis and computational simulations contained herein are free of error—however, errors may remain and the reader is cautioned against drawing conclusions from this analysis without performing due diligence. Under no circumstances does BlockScience warrant the veracity or accuracy of the analysis, and the reader is encouraged to perform their own analysis.

Preface

Notation

This section introduces some basic notation that is used to describe sets, functions, different spaces and how they are manipulated.[1]

1. **Set**: "X"

   a collection of things is represented by an upper-case letter, such as X

2. **Element**: "x"

   a thing in a set is represented by a lower-case letter, such as x

3. **Membership**: "$\in$"

   an element x *that is in a set* X *is represented by* $x \in X$

4. **Cardinality**: "$|X|$"

   the number of elements in a set X *is represented by its cardinality* $|X|$

5. **Product**: "$\times$"

   an ordered pair of elements $x \in X$ *and* $y \in Y$ *is represented by a tuple* $(x, y) \in X \times Y$

6. **Subset**: "$\subseteq$"

   a set Z *being a collection of elements of a set* Y *is represented by* $Z \subseteq Y$

7. **Sequence**: "$\{\cdot\}$"

   a collection of elements with an increasing integer index $k = 1, 2, \ldots, K$ *is represented by* $\{x_k\}_{k=1}^K$*, where* K *may equal infinity*

8. **Convergence**: "$\rightarrow$"

   the convergence of a sequence of values $\{x_k\}_{k=1}^K$ *of a variable* $x \in X$ *to a value* $x^\star \in X$ *(its "limit"), each* $x_k \in X$*, is represented by* $x \rightarrow x^\star$

9. **Relation**: "R"

   a subset R *of a product* $X \times Y$ *is represented as* xRy *for* $(x, y) \in R \subseteq X \times Y$

[1] This section provides just enough detail to 'hit the ground running'—for a comprehensive and very readable exposition of set theory see e.g. Ok, Efe A., Real Analysis with Economic Applications, Princeton UP, 2007.

10. **Function**: "f"

    a relation f that 1) associates to every $x \in X$ a $y \in Y$ via xfy, and 2) relates xfy and xfz, $x \in X$, $y, z \in Y$ only when $y = z$, is represented by $y = f(x)$

11. **Implication**: "$\Rightarrow$"

    if statement A implies statement B, this may be represented by $A \Rightarrow B$

12. **Equivalence**: "$\Leftrightarrow$"

    if statement A implies statement B and statement B implies statement A, this may be represented by $A \Leftrightarrow B$ or $B \Leftrightarrow A$

13. **Identically Equal**: "$\equiv$"

    if variable x is identically equal to variable y, this may be represented by $x \equiv y$

14. **Summation**: "$\sum$"

    a sum $x_1 + x_2 + \ldots + x_K$ of K elements, where K may equal infinity, is represented by $\sum_{k=1}^{K} x_k$

15. **Euclidean Space**: "$\mathbb{R}$", "$\mathbb{R}_+$", "$\mathbb{R}_{++}$", "$\mathbb{R}^N$", "$\mathbb{R}_+^N$", "$\mathbb{R}_{++}^N$"

    respectively the real line, non-negative real line, strictly positive real line, N-dimensional Euclidean space, N-dimensional Euclidean space with non-negative orthant, and N-dimensional Euclidean space with strictly positive orthant, where N is a finite integer

16. **Natural Numbers**: "$\mathbb{N}_0$"

    the set of integers including zero, i.e. $0, 1, 2, \ldots$

Formatting

Cross-reference hyperlinks are used throughout the documentation to facilitate fast lookup and traversal, and are color-coded for ease of use. The following cross-reference types and associated colors are included:

- **Internal sectioning and numbering**
- **External URL hyperlinks**

1 Executive Summary

This Report presents an analysis by BlockScience of the Hydra **Omnipool** Automated Market Maker (AMM) as of early March 2022, providing an overview of the AMM specification and its mechanisms, and discussing economic implications of those mechanisms for liquidity provision and trading activity. The analysis concludes that the **Omnipool**'s mechanisms for adding and removing asset liquidity and conducting trading operations between assets (including the protocol—or 'pool'—token '`LRNA`', although at the time of this writing the purchase of `LRNA` from the **Omnipool** will not be an accessible mechanism to a trader) function as intended and provide the stipulated properties from which conclusions regarding performance and efficiency metrics such as impermanent loss[2] and slippage may be derived. Of note is the tracking within the **Omnipool** of the distortionary effects of `LRNA` minting on prices, and the novel mechanism for utilizing fees from `LRNA` trading to offset these effects and, at the same time, increase the value of the ecosystem's governance token, `HDX`, in the **Omnipool**.

The **Omnipool** is introduced in Section 2 and its specification is provided in Section 3, while Section 4 treats pool viability (without assumptions on agent behavior or associated 'tokenomics') and the economics of liquidity provision impermanent loss and trading price slippage.

2 Introduction

The HydraDX **Omnipool** automated market maker (AMM) provides single-asset exposure and impermanent loss mitigation using an elastic supply mechanism for its protocol—or 'pool'—token, '`LRNA`', supporting a single pool of all assets. The **Omnipool** incentivizes liquidity providers to commit liquidity in single assets via returns from trading fees. At the time of liquidity provision the **Omnipool** shares in the liquidity provider's exposure to external asset price changes by minting `LRNA` of the same value as the provided liquidity, providing a measure of impermanent loss mitigation that is resolved when the position is unwound. As with all AMMs, pool liquidity is used to ensure that traders can participate in price discovery through trading (swap) mechanisms. The **Omnipool** builds upon the well-understood Constant Function Market Maker (CFMM) invariant property approach, applying this approach to 'subpools' comprised of an asset and a quantity of `LRNA` associated with that asset.

Although in its nascent stages, the **Omnipool** is nevertheless in a form (as communicated by Hy-

[2] A newer and perhaps more appropriate term for this phenomenon is 'divergence loss', but in the interest of readability in connection with previous and extant research the term 'impermanent loss' is used throughout this Report.

draDX between January and early March of 2022) that is robust enough to admit an analysis of its specification and general economic conclusions regarding its viability (up to the 'tokenomics' of the system, as behavioral assumptions are not made in this Report). Novel in the specification of the **Omnipool** is the usage of an accounting 'ledger' that keeps track of the extent to which the minting of `LRNA` distorts prices, and of a mechanism that uses fees accrued from `LRNA` operations to offset this distortion. Provided that distortionary effects are fully offset, any excess fees remaining are used to increase the value of the **Omnipool**'s holding of the ecosystem's governance token, `HDX`. This has the effect of directly connecting increased usage of the **Omnipool** for its primary trading function on the one hand, and the value of the ecosystem's `HDX` token as a reflection of this increased usage on the other.

The analysis in this Report concludes that the **Omnipool**'s associated mechanisms for adding and removing asset liquidity and conducting trading operations between assets (including the protocol asset `LRNA`, although at the time of this writing the purchase of `LRNA` from the **Omnipool** will not be an accessible mechanism to a trader) function as intended—they preserve pool value for assets uninvolved in liquidity and/or trade events, while ensuring that invariant properties, such as the CFMM for subpools,[3] are preserved. The mechanisms also provide the stipulated properties from which conclusions regarding performance and efficiency metrics such as impermanent loss and slippage may be derived, particularly as the **Omnipool** engages in impermanent loss mitigation by taking a risk position with every provision of liquidity to the pool.

The structure of this Report is as follows. In Section 3 the **Omnipool** specification is presented, commencing with the global and local state space representations (Section 3.1), proceeding to overall system constraints (Section 3.2) and concluding with an in-depth description of the mechanisms for liquidity and trading operations (Section 3.3).

Section 4 then covers the economic implications of the **Omnipool** specification (without, as mentioned, delving into the tokenomics), covering pool viability (Section 4.1), impermanent loss (Section 4.2) and slippage (Section 4.3).

[3] It is noted that this 'subpool' approach is similar to, but distinct from, the Bancor 2.1 specification.

3 The Omnipool Specification

The **Omnipool** specification is presented using the *state space* approach—the system state variables are explicitly defined, and their constraints and relationships due to the **Omnipool**'s specified mechanisms are motivated and presented.

3.1 State Space Representation

As the system is comprised of both the **Omnipool** and its users, *global* (in the sense of "pool level") and *local* (in the sense of "participant/agent level") states are each treated in what follows. In addition, global parameters (such as fees) and derived metrics (such as prices) are defined, covering variables that may be conditioned upon to define other metrics such as impermanent loss and slippage. The following definitions in the next two subsections, while terse in their presentation, provide a useful series of 'lookup tables' when consulting the mechanisms and economic implications later in the Report.

3.1.1 Global state (System Level)

Omnipool State		
Symbol	**Description**	**Domain**
N	Number of (non-`LRNA`, non-`HDX`) assets in the **Omnipool**	$\mathbb{N}_0$
R_i	Quantity of (non-`LRNA`, non-`HDX`) asset i	$\mathbb{R}_{++}$
$\mathbf{R}$	Vector of (non-`LRNA`, non-`HDX`) asset quantities	$\mathbb{R}_{++}^N$
Q_i	Quantity of `LRNA` matching asset i	$\mathbb{R}_{++}$
$\mathbf{Q}$	Vector of (non-`HDX` associated) `LRNA` quantities	$\mathbb{R}_{++}^N$
R_{HDX}	Quantity of `HDX`	$\mathbb{R}_{++}$
Q_{HDX}	Quantity of `LRNA` associated with `HDX`	$\mathbb{R}_{++}$
S_i	Total quantity of pool shares outstanding for asset i	$\mathbb{R}_{++}$
$\mathbf{S}$	Vector of total pool share quantities outstanding	$\mathbb{R}_{++}^N$
B_i	Quantity of pool shares for asset i owned by the protocol	$\mathbb{R}_{+}$
$\mathbf{B}$	Vector of pool shares owned by the protocol	$\mathbb{R}_{+}^N$
L	`LRNA` imbalance ledger	$\mathbb{R}$

Table 1: Global States, Pool

Global Parameters		
Symbol	**Description**	**Domain**
f_Q	`LRNA` (protocol) fee	$[0, 1)$
f_R	Asset fee	$[0, 1)$

Table 2: Parameters

Global Stateful Metrics		
Symbol	**Description**	**Relation**
Q	Total `LRNA` in the **Omnipool** allocated to assets	$\sum_i Q_i + Q_{HDX}$
p_i^Q	Price of asset i in `LRNA`	$\frac{Q_i}{R_i}$
$\mathbf{p}^Q$	Vector of asset prices in `LRNA`	$(p_1^Q, \ldots, p_N^Q)$
p_i^j	Price of asset i denominated in asset j	$\frac{Q_i}{Q_j} \frac{R_j}{R_i}$
$\mathbf{p}^j$	Vector of asset prices denominated in asset j	$(p_1^j, \ldots, p_N^j)$
Y_i	Swap invariant of asset i in the **Omnipool**	$Q_i R_i$

Table 3: Metrics

3.1.2 Local State (Agent Level)

Agent State		
Symbol	**Description**	**Domain**
α	Index of agent (trader or liquidity provider)	$\mathbb{N}_0$
$i(\alpha)$	Index of a single asset associated with α	$\{1, \ldots, N\}$
r_i^α	Quantity of asset $i(\alpha)$ held by α	$\mathbb{R}_+$
$\mathbf{r}^\alpha$	Vector of assets held by α	$\mathbb{R}_+^N$
q^α	Quantity of `LRNA` held by α	$\mathbb{R}_+$
s_i^α	Quantity of $i(\alpha)$ pool shares held by α	$\mathbb{R}_+$
$\mathbf{s}^\alpha$	Vector of pool shares held by α	$\mathbb{R}_+^N$
p_i^α	Price of asset $i(\alpha)$ at the time of liquidity provision[4] by α	$\mathbb{R}_{++}^2$

Table 4: Local States, Agent

3.2 System Constraints

3.2.1 Total pool shares

The quantity of pool shares in the system, S_i, is always the sum of pool shares held by the protocol and pool shares held across all agents:

$$S_i \equiv B_i + \sum_\alpha s_i^\alpha.$$

3.3 Mechanisms

For any variable X, we will use the notation $X^+ = X + \Delta X$ (note that we are allowing ΔX to be negative). For a *pool* or *protocol* variable, such as the change in the **Omnipool**'s asset i balance ΔR_i or the change in the protocol's `LRNA` imbalance ledger ΔL, its sign will indicate the change from the perspective of the pool or the protocol (e.g. $\Delta R_i > 0$ indicates the pool balance R_i has increased). By contrast, for an *agent* variable, such as the change in the agent's holding of asset i, Δr_i^α, its sign will indicate the change from the perspective of the agent (e.g. $\Delta r_i^\alpha > 0$ indicates the agent's holdings of asset i, r_i^α, has increased). This is important to keep in mind when evaluating the following mechanisms—for example, the add liquidity event treated in Section 3.3.1 below has an agent depositing a positive amount $\Delta R_i > 0$ into the pool, which corresponds to a decline in their asset i holding by a negative amount $\Delta r_i^\alpha < 0$. Thus, $\Delta r_i^\alpha = -\Delta R_i$ for this event.

3.3.1 Add Asset Liquidity (single asset, existing token)

The *add liquidity* mechanism (cf. AddLiquidity.ipynb[5]) has the generalized form

$$(\mathbf{R}^+, \mathbf{S}^+, \mathbf{Q}^+, \mathbf{r}^{\alpha+}, \mathbf{s}^{\alpha+}) = f_{liq+}(\Delta r_i^\alpha, \mathbf{R}, \mathbf{S}, \mathbf{Q}, \mathbf{r}^\alpha, \mathbf{s}^\alpha)$$

where an agent indexed by $\alpha \in 1, 2, \ldots$ provides an amount Δr_i^α of an asset i to the pool.[6] As a consequence, an amount of `LRNA` is minted and added to the pool. In return for providing liquidity,

[4] The **Omnipool** treats each liquidity provision event as a separate position–hence, each agent only has a single liquidity event associated with their index (naturally the same liquidity provider may be associated with multiple indices).

[5] Here and in the following mechanisms links to associated python Jupyter notebooks have been provided as they existed at the time of this writing: an entry point into the associated Github repository is `https://github.com/galacticcouncil/HydraDX-simulations/blob/main/README.md`. These links may become stale over time, and the reader is referred to the HydraDX Documentation site for links to the most current **Omnipool** documentation, description etc.

[6] Note that here and in what follows, a mechanism function f, such as f_{liq+}, is described in detail, and the state updates are explicitly mentioned in each mechanism subsection.

an amount of pool shares for asset i is minted and allocated to the agent α. The impact of the newly minted LRNA on the LRNA imbalance ledger is accounted for by increasing this imbalance.

Requirements

1. Adding liquidity must leave prices (in terms of LRNA) p_j^Q unchanged for all assets $j = 1, \ldots, N$.

2. Adding liquidity must leave the ratio $\frac{R_i}{S_i}$ unchanged—this is the value of one pool share for asset i in terms of asset i.

3. Adding liquidity must leave the ratio of LRNA accounted as an imbalance to total LRNA, $\frac{L}{Q}$, unchanged. The LRNA imbalance ledger $L < 0$ keeps track of the 'excess' LRNA that is in the **Omnipool** as a whole, stemming largely from swap operations (cf. Section 3.3.4) and the adding of liquidity. In the latter case, an existing non-zero imbalance L indicates that there is already an excess of LRNA, and so when the protocol mints LRNA at the same value as the liquidity provided (ΔQ_i above) this *exacerbates* the imbalance, which must be accounted for.

   To accomplish this, note that an increase in LRNA ΔQ_i is an increase in the **Omnipool**'s balance of LRNA as a whole, Q. Given a level of imbalance L prior to liquidity provision, the fraction of LRNA in the **Omnipool** that is accounted for by L is just the ratio L/Q. In order to account for the increase in LRNA from ΔQ_i, then, the value of L is adjusted in order to keep this ratio unchanged:

$$\frac{L^+}{Q^+} = \frac{L + \Delta L}{Q + \Delta Q} = \frac{L}{Q},$$

   or, since $\Delta Q = \Delta Q_i$,

$$\frac{L + \Delta L}{Q + \Delta Q_i} = \frac{L}{Q} \Rightarrow \Delta L = L \frac{\Delta Q_i}{Q}. \tag{3.1}$$

   Note that $\Delta L < 0$ since $L < 0$, indicating that the LRNA imbalance is increased following the minting of LRNA ΔQ_i. This increase may also be noted in an equivalent formulation of the above,

$$\Delta L = \frac{Q_i}{R_i} \frac{L}{Q} \Delta R_i = p_i^Q \frac{L}{Q} \Delta R_i, \tag{3.2}$$

   which indicates that ΔL may also be seen as an adjustment taking into consideration the impact of the value of liquidity provision $p_i^Q \Delta R_i$ in LRNA on the existing LRNA imbalance.[7]

[7] This relation may also be derived by finding the LRNA imbalance adjustment required to keep a 'target' spot price $p_i^Q(1 + L/Q)$ constant—it is straightforward to show that the two values of ΔL in equations (3.1) and (3.2) are the same.

Constraints

1. The change in agent's α holdings of the asset is negative:

$$\Delta r_i^\alpha < 0.$$

2. The agent α is endowed with at least the amount of the asset to be provided:

$$r_i^\alpha \geq -\Delta r_i^\alpha.$$

State update (global)

1. The change in the pool quantity of asset i, $\Delta R_i > 0$, equals the amount added:

$$\Delta R_i = -\Delta r_i^\alpha > 0, R_i^+ = R_i + \Delta R_i.$$

2. The quantity of `LRNA` allocated to asset i minted by the protocol, $\Delta Q_i > 0$, is:

$$\Delta Q_i := Q_i \frac{\Delta R_i}{R_i} > 0, Q_i^+ = Q_i + \Delta Q_i.$$

   This preserves the price p_i^Q of asset i in terms of `LRNA`, since

$$p_i^Q := \frac{Q_i}{R_i}.$$

3. The quantity of pool shares for asset i minted by the protocol, $\Delta S_i > 0$, is:

$$\Delta S_i = S_i \frac{\Delta R_i}{R_i} > 0, S_i^+ = S_i + \Delta S_i.$$

4. The change in the `LRNA` imbalance ledger, $\Delta L < 0$, is the adjustment required to keep the ratio of the imbalance to the total `LRNA` in the **Omnipool** constant following the addition of $\Delta Q_i > 0$ to the pool:

$$\Delta L = \Delta Q_i \frac{L}{Q},\ L^+ = L + \Delta L.$$

State update (local)

1. The quantity of asset i held by the agent declines in the amount $\Delta r_i^\alpha < 0$ provided to the **Omnipool**:

$$r_i^{\alpha+} = r_i^\alpha + \Delta r_i^\alpha.$$

2. The quantity of pool shares distributed to agent α, $\Delta s_i^\alpha > 0$, equals the amount of new shares minted:

$$\Delta s_i^\alpha = \Delta S_i > 0, s_i^{\alpha+} = s_i^\alpha + \Delta s_i^\alpha.$$

3. The price associated with this liquidity position for agent α, p_i^α, is defined as the price of the asset i denominated in Q at the time of the provision, p_i^Q:

$$p_i^\alpha := p_i^Q.$$

3.3.2 Add Asset Liquidity (single asset, new token)

The add new token liquidity mechanism (cf. AddToken.ipynb) has the generalized form

$$(\mathbf{R}^+, \mathbf{S}^+, \mathbf{Q}^+, \mathbf{B}^+, N^+) = f_{tok+}(\Delta R_{N+1}, \mathbf{R}, \mathbf{S}, \mathbf{Q}, \mathbf{B}, N, \hat{p}_{N+1}^Q),$$

where the protocol adds a quantity $\Delta R_{N+1} > 0$ of risk asset $N+1$ to the **Omnipool**. The protocol then receives the new pool shares for that risk asset, and mints an amount of `LRNA` corresponding to the value of ΔR_{N+1} when provided. The impact of the newly minted `LRNA` on the `LRNA` imbalance ledger is accounted for by increasing this imbalance.

Central to the addition of a completely new token is the price at which the token is valued, $\hat{p}_{N+1}^Q$. This is assumed to be provided to the protocol exogenously, perhaps as the result of a governance process that sets this price.

Requirements

1. Adding liquidity in a new token must leave prices (in terms of `LRNA`) p_j^Q unchanged for all assets $j = 1, \ldots, N$;

2. The amount of `LRNA` created for asset $N+1$, Q_{N+1}, must equal the value of ΔR_{N+1} at the

exogenously-provided price $\hat{p}^Q_{N+1}$:

$$Q_{N+1} := \hat{p}^Q_{N+1} \Delta R_{N+1}.$$

3. As with adding liquidity for an existing asset in the **Omnipool** (cf. Section 3.3.1), adding liquidity for a completely new asset must also leave the ratio of `LRNA` accounted as an imbalance to total `LRNA`, $\frac{L}{Q}$, unchanged.

State update (global)

1. The total quantity of asset $N+1$ in the pool, $R_{N+1} > 0$, equals the amount added by the protocol:
$$R_{N+1} = \Delta R_{N+1} > 0.$$

2. The initial quantity of total pool shares for asset $N+1$, $S_{N+1} > 0$, is set to R_{N+1} (this is arbitrary and acts as a normalization only):
$$S_{N+1} = R_{N+1}.$$

3. The initial quantity of pool shares owned by the protocol for asset $N+1$, $B_{N+1} > 0$, is set to S_{N+1}:
$$B_{N+1} = S_{N+1} = R_{N+1}.$$

4. The initial quantity of `LRNA` allocated to asset $N+1$, $Q_{N+1} > 0$, is:
$$Q_{N+1} = \hat{p}^Q_{N+1} R_{N+1}.$$

5. The number of assets in the pool is increased: $N^+ = N+1$.

6. The change in the `LRNA` imbalance ledger, $\Delta L < 0$, is the adjustment required to keep the ratio of the imbalance to the total `LRNA` in the **Omnipool** constant following the creation of $Q_{N+1} > 0$:
$$\Delta L = Q_{N+1} \frac{L}{Q}, \; L^+ = L + \Delta L.$$

3.3.3 Withdraw Asset Liquidity (single asset)

The withdraw liquidity mechanism (cf. WithdrawLiquidity.ipynb) has the generalized form

$$(\mathbf{R}^+, \mathbf{S}^+, \mathbf{Q}^+, \mathbf{B}^+, N^+, \mathbf{r}^{\alpha+}, \mathbf{s}^{\alpha+}) = f_{liq-}(\Delta s_i^\alpha, \mathbf{R}, \mathbf{S}, \mathbf{Q}, \mathbf{B}, N, \mathbf{r}^\alpha, \mathbf{s}^\alpha)$$

where an agent α returns a quantity Δs_i^α of asset i's pool shares to the protocol. In return, a quantity Δr_i^α of asset i and a quantity Δq_i^α of `LRNA` (which may be zero) are provided to the agent, the amount of `LRNA` corresponding to asset i is reduced, and the total pool shares for asset i are reduced. The impact of the removed `LRNA` on the `LRNA` imbalance ledger is accounted for by decreasing this imbalance.

Requirements

1. Withdrawing liquidity must leave prices (in terms of `LRNA`) p_j^Q unchanged for all assets $j = 1, \ldots, N$.
2. Withdrawing liquidity must leave the ratio $\frac{R_i}{S_i}$ unchanged—this is the value of one pool share for asset i in terms of asset i.
3. When withdrawing liquidity, the value of the assets withdrawn must conform to the **Omnipool**'s impermanent loss (IL) mitigation specification. The total initial pool value of the asset i in terms of `LRNA`, $2p_i^\alpha \Delta R_{i0}$, (where ΔR_{i0} was the initial asset amount contributed), is augmented or diminished by changes in the price of `LRNA`, i.e. in terms of a factor p_i^Q / p_i^α. This yields the pool value of the contribution to asset i, V_i, at the time of withdrawal:

   $$V_i = 2\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2} p_i^\alpha \Delta R_{i0}.$$

   Part of the value of the pool from agent α's liquidity provision is attributed to the agent, while part is attributed to the pool itself. These parts are allocated in the ratio of $p_i^Q / p_i^\alpha : 1$ to the agent and the pool, respectively. This means that the value attributed to the agent and available for withdrawal will be

   $$V_i^\alpha = \frac{p_i^Q / p_i^\alpha}{1 + p_i^Q / p_i^\alpha} V_i = \frac{p_i^Q / p_i^\alpha}{1 + p_i^Q / p_i^\alpha} 2\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2} p_i^\alpha \Delta R_{i0}. \tag{3.3}$$

This may be rewritten as

$$V_i^\alpha = \frac{2p_i^Q p_i^\alpha}{p_i^Q + p_i^\alpha}\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2}\Delta R_{i0}. \tag{3.4}$$

To return this value to the agent requires deriving ΔR_{i0}, the amount of asset i initially contributed to the pool by agent α. To find this value, first consider the amount of reserve asset i, ΔR_i^α, that corresponds to their share holding Δs_i^α:

$$\Delta R_i^\alpha = -\frac{\Delta s_i^\alpha}{S_i} R_i, \tag{3.5}$$

where $\Delta R_i^\alpha > 0$ is from the perspective of the agent.

Any change in the amount of asset i in the reserve corresponds to a change in its price, leading to a reserve amount increasing by a factor of $\left(\frac{p_i^\alpha}{p_i^Q}\right)^{1/2}$ as prices change from p_i^α to p_i^Q. Thus the initial reserve contribution ΔR_{i0} is related to ΔR_i^α by

$$\Delta R_{i0} = \left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2}\Delta R_i^\alpha. \tag{3.6}$$

Putting together equations (3.4) - (3.6) yields the value provided to the agent in terms of the current price p_i^Q, the price p_i^α at the time of contribution, and the pool state at the time of withdrawal, given the pool shares they wish to redeem:

$$V_i^\alpha = -\frac{2\left(p_i^Q\right)^2}{p_i^Q + p_i^\alpha}\frac{\Delta s_i^\alpha}{S_i} R_i. \tag{3.7}$$

The agent receives both $\Delta r_i^\alpha = -\Delta R_i > 0$ and possibly $\Delta q_i^\alpha \geq 0$, the total value of which must equal V_i^α. This relation is sufficient to derive the quantity of `LRNA` which may be necessary to provide to the agent, if the value of reserve distributed is less than V_i^α:

$$p_i^Q \Delta r_i^\alpha + \Delta q_i^\alpha = V_i^\alpha \Rightarrow \Delta q_i^\alpha = p_i^Q\left(-\frac{2p_i^Q}{p_i^Q + p_i^\alpha}\frac{\Delta s_i^\alpha}{S_i}R_i + \Delta R_i\right). \tag{3.8}$$

4. Withdrawing liquidity must leave the ratio of `LRNA` accounted as an imbalance to total `LRNA`, i.e. $\frac{L}{Q}$, unchanged. Recall that L keeps track of the ‘excess’ `LRNA` that is in the **Omnipool** as a whole, stemming largely from swap operations (cf. Section 3.3.4) and the adding of liquidity. When

liquidity is withdrawn, this imbalance is (partially or wholly) rectified as LRNA is removed from the system in the amount $\Delta Q_i < 0$. This must be accounted for by adjusting the LRNA imbalance amount L accordingly.

A similar argument to that given for the add liquidity mechanism (cf. Section 3.3.1) indicates that the form of $\Delta L > 0$ is unchanged, but its sign is reversed–the imbalance accounted for in the ledger L is *reduced* by an amount:

$$\Delta L = L\frac{\Delta Q_i}{Q} = p_i^Q \frac{L}{Q}\Delta R_i,$$

as indicated in equations (3.1) and (3.2), respectively. Thus, a liquidity withdrawal event helps to 'redress the balance' of excess LRNA when the amount ΔQ_i is removed from the **Omnipool**, while ensuring that the ratio of the imbalance amount to the total amount of LRNA in the **Omnipool** remains unchanged.

Constraints

1. The change in the agent's holdings of pool shares is negative:

$$\Delta s_i^\alpha < 0.$$

2. The agent can only redeem up to the total number of pool shares of an asset held:

$$s_i^\alpha \geq -\Delta s_i^\alpha.$$

3. The **Omnipool**'s holding of pool shares $\mathbf{B}$ is non-negative for each asset: if the transfer to the agent of LRNA, Δq_i^α, is zero, then it must be from (3.8) that:

$$\frac{\Delta R_i}{R_i} = \frac{2p_i^Q}{p_i^Q + p_i^\alpha}\frac{\Delta s_i^\alpha}{S_i}.$$

Combining this with (3.5) and the definition of the total supply of pool shares S_i for asset i,

$$\Delta S_i = \Delta s_i^\alpha + \Delta B_i$$

yields

$$\frac{\Delta R_i}{R_i} = \frac{\Delta S_i}{S_i} = \frac{\Delta s_i^\alpha + \Delta B_i}{S_i} = \frac{2p_i^Q}{p_i^Q + p_i^\alpha}\frac{\Delta s_i^\alpha}{S_i} \Rightarrow$$

$$\Delta B_i = \left(\frac{2p_i^Q}{p_i^Q + p_i^\alpha} - 1\right)\Delta s_i^\alpha = \frac{p_i^Q - p_i^\alpha}{p_i^Q + p_i^\alpha}\Delta s_i^\alpha. \tag{3.9}$$

Notice that the sign of ΔB_i depends upon $p_i^Q - p^\alpha$: an increase in the price of the reserve would necessitate a distribution of pool shares from the protocol ($\Delta B_i < 0$), while a decrease in the price would necessitate a transfer of pool shares from the agent to the protocol ($\Delta B_i > 0$). The current version of the protocol is designed to gradually accumulate a liquidity position as agents add liquidity—for this reason, if it should be the case that $\frac{p_i^Q - p_i^\alpha}{p_i^Q + p_i^\alpha}\Delta s_i^\alpha < 0$, i.e. (cf. eqn. 3.9) if $p_i^Q > p_i^\alpha$, then ΔB_i is set to zero. The deficit in value to be transferred to the agent is then made up by the transfer of Δq_i^α in `LRNA`.

Putting these restrictions on ΔB_i together, we have that

$$\Delta B_i = \max\left\{\frac{p_i^Q - p_i^\alpha}{p_i^Q + p_i^\alpha}\Delta s_i^\alpha, 0\right\} \geq 0.$$

The motivation for this mechanism is the risk-sharing taking place between the agent and the protocol when liquidity is added (cf. Section 3.3.1): when an agent provides liquidity to the protocol, the protocol also assumes the risk of impermanent loss when it mints `LRNA` equal in value to the value of the reserve asset provided at the time of provision. This risk is realized when the agent withdraws liquidity, and the (implicit) pool shares in asset i held by the protocol are realized.

For example, if upon withdrawal the current price of asset i in terms of `LRNA`, p_i^Q, is greater than the price at the time of liquidity provision, p_i^α, then

$$p_i^Q > p_i^\alpha \Leftrightarrow \frac{Q_i}{R_i} > \frac{Q_{i0}}{R_{i0}} \Rightarrow \frac{Q_i}{Q_{i0}} > \frac{R_i}{R_{i0}}.$$

Thus, the amount of `LRNA` has grown over the duration of liquidity provision by more than the corresponding growth of reserve i, and $\Delta q_i^\alpha > 0$ in `LRNA` must be provided in addition to $\Delta r_i^\alpha = \Delta R_i$ to the agent. We then have that $\Delta S_i = \Delta s_i^\alpha < 0$.

By contrast, suppose that during the duration of liquidity provision the price of the asset in terms of LRNA has fallen, i.e. $p_i^Q < p_i^\alpha$. Then

$$\frac{Q_i}{Q_{i0}} < \frac{R_i}{R_{i0}}$$

and reserve asset i has grown by more than the corresponding growth of LRNA. Then $\frac{p_i^Q - p_i^\alpha}{p_i^Q + p_i^\alpha}\Delta s_i^\alpha > 0$ and ΔB_i will be *positive*. Since

$$\Delta S_i = \Delta s_i^\alpha + \Delta B_i,$$

$\Delta B_i > 0$ may be interpreted as a *transfer* of pool shares to the protocol, to cover the decline in value of the asset. Specifically,

$$\Delta S_i = \Delta s_i^\alpha + \frac{p_i^Q - p_i^\alpha}{p_i^Q + p_i^\alpha}\Delta s_i^\alpha = \Delta s_i^\alpha \left(1 + \frac{p_i^Q - p_i^\alpha}{p_i^Q + p_i^\alpha}\right) = \Delta s_i^\alpha \frac{2p_i^Q}{p_i^Q + p_i^\alpha},$$

i.e. $\Delta s_i^\alpha < \Delta S_i < 0$. This means that the amount of pool shares actually redeemed to provide ΔR_i is *smaller* than the amount of shares provided by the agent (recall both Δs_i^α and ΔS_i are *negative*), with the residual transferred to the protocol in the form of ΔB_i.

State update (global)

1. The change in the pool shares held by the protocol, $\Delta B_i \geq 0$, is (as derived above):

$$\Delta B_i = \max\left\{\frac{p_i^Q - p_i^\alpha}{p_i^Q + p_i^\alpha}\Delta s_i^\alpha, 0\right\},\ B_i^+ = B_i + \Delta B_i.$$

2. The change in the pool shares for asset i, $\Delta S_i < 0$, equals the sum of the shares submitted by the agent, Δs_i^α and the shares submitted by the protocol, ΔB_i:

$$\Delta S_i = \Delta s_i^\alpha + \Delta B_i,\ S_i^+ = S_i + \Delta S_i.$$

3. The change in the pool quantity of asset i, $R_i < 0$, is such that the fraction of reserve distributed corresponds to the fraction of total pool shares S_i burnt:

$$\Delta R_i = R_i \frac{\Delta S_i}{S_i},\ R_i^+ = R_i + \Delta R_i.$$

4. The change in the pool quantity of LRNA for asset i, $\Delta Q_i < 0$, is the change in the value of the

reserve ΔR_i withdrawn from the pool:

$$\Delta Q_i = p_i^Q \Delta R_i,\ Q_i^+ = Q_i + \Delta Q_i.$$

We note that because $p_i^Q = Q_i/R_i$, we may also write

$$\Delta Q_i = Q_i \frac{\Delta R_i}{R_i}.$$

5. If $R_i^+ = 0$, i.e. all liquidity in asset i has been removed, then it is straightforward to conclude that $S_i^+ = 0$ and $Q_i^+ = 0$, and consequently $N^+ = N - 1$. Otherwise, if $R_i^+ > 0$, then $N^+ = N$.

6. The change in the `LRNA` imbalance ledger, $\Delta L > 0$, is the adjustment required to keep the ratio of the imbalance to the total `LRNA` in the **Omnipool** constant following the removal of $\Delta Q_i < 0$ from the pool:

$$\Delta L = \Delta Q_i \frac{L}{Q},\ L^+ = L + \Delta L.$$

State update (local)

1. The quantity of asset i reserve distributed to agent α, $\Delta r_i^\alpha > 0$, equals the amount of the reserve removed from the pool:

$$\Delta r_i^\alpha = -\Delta R_i,\ r_i^{\alpha+} = r_i^\alpha + \Delta r_i^\alpha.$$

2. The quantity of `LRNA` distributed to agent α, $\Delta q^\alpha \geq 0$, is the amount distributed as the residual of the value V_i^α the agent is entitled to, and the value of the reserve distribution:

$$\Delta q^\alpha = \Delta q_i^\alpha = V_i^\alpha - p_i^Q \Delta r_i^\alpha,\ q^{\alpha+} = q^\alpha + \Delta q^\alpha.$$

We note in passing that when $\Delta B_i = 0$, we have:

$$\Delta q^\alpha = -p_i^Q \left(\frac{2p_i^Q}{p_i^Q + p_i^\alpha} \frac{\Delta S_i}{S_i} R_i + \Delta r_i^\alpha \right),$$

otherwise $\Delta q^\alpha = 0$.

3.3.4 Swapping LRNA with asset i

The *swap* LRNA with asset i and *swap* asset i with LRNA mechanisms[8] (cf. SwapLRNA.ipynb) both have the generalized form

$$(\mathbf{R}^+, \mathbf{Q}^+, \mathbf{r}^{\alpha+}, q^{\alpha+}) = f_{swap}(\Delta r_i^\alpha, \Delta q^\alpha, \mathbf{R}, \mathbf{Q}, \mathbf{r}^\alpha, q^\alpha; f_Q, f_R),$$

where

- an agent α (a "trader") may sell either a specified quantity $\Delta q^\alpha < 0$ of LRNA to the **Omnipool**, or buy a specified quantity Δr_i^α of the asset from the **Omnipool**;
- the **Omnipool** protocol itself may sell a quantity $\Delta R_i > 0$ to the **Omnipool** in return for LRNA, as part of the asset-for-asset swap (cf. 3.3.5).

When LRNA is swapped into the pool in return for an asset, an *asset fee* f_R is assessed on the asset amount, and an amount $\Delta r_i^\alpha > 0$ net of this fee is returned to the trader. If the protocol is buying LRNA, then when an asset is swapped into the pool a *protocol fee* f_Q is assessed on the LRNA amount and an amount $\Delta q^\alpha > 0$ of LRNA is provided, net of fees. (Either or both of these fees may be zero, depending upon the **Omnipool**'s implementation and eventual governance structure for adjusting the fee parameters.)

Requirement

A swap keeps the invariant for asset i, $Y_i := R_i Q_i$, constant during the transaction. Note that when there is a protocol or asset fee assessed for the swap, the value of Y_i will generally change after the swap is conducted. This is the underlying invariant used, for example, in a Constant Function Market Maker (CFMM), applied here to the 'subpool' comprised of asset i's pool balance and its associated balance of LRNA.

Constraints

When LRNA is sold into the **Omnipool**:

[8] As of this writing, the asset-for-LRNA mechanism is **inaccessible** to a trader, but its specification here is useful for the formal description of an asset-for-asset swap described later in Section 3.3.5.

(a) the change in agent's α holdings of LRNA is negative:

$$\Delta q^\alpha < 0;$$

(b) the trader must have at least the amount Δq_i^α to trade:

$$q^\alpha \geq -\Delta q^\alpha.$$

State updates (global)

1. When LRNA is sold into the **Omnipool** at a specified quantity $\Delta q^\alpha > 0$:

(a) the quantity of asset i returned by the mechanism (including fees), $\Delta R_i < 0$, is:

$$\Delta R_i := -R_i \frac{\Delta q^\alpha}{Q_i + \Delta q^\alpha} < 0; \tag{3.10}$$

the transactions fee $f_R \in [0, 1)$ is applied to ΔR_i, and the total fee amount is added to the quantity of asset i. The new quantity of asset i after the swap, then, is:

$$R_i^+ = R_i + (1 - f_R)\Delta R_i;$$

(b) a *ledger update* is performed to update the value of the LRNA imbalance state variable L. The LRNA imbalance is an accounting for the deviation in LRNA that is induced when a swap changes the price of asset i. If the reserve and LRNA amounts before the swap are R_i and Q_i, respectively, the price of asset i at that time is

$$p_i^Q := \frac{Q_i}{R_i}.$$

When the swap occurs with an amount $\Delta Q_i > 0$ swapped into the **Omnipool** and $\Delta R_i < 0$ generated by the swap mechanism, the price changes to

$$p_i^{Q+} = \frac{Q_i^+}{R_i^+},$$

with R_i^+ as above. The goal is to find that amount of LRNA $\Delta L < 0$ that would have been necessary to remove from Q_i, in order to keep prices *constant*—in other words, the net change in `LRNA` $\Delta Q_i + \Delta L$ is that amount of `LRNA` required to treat the swap *as if* the removal ΔR_i were a liquidity withdrawal event. While the actual price does change as a result of the swap, accumulations in (the negative balance) L indicate how much `LRNA` has been accumulated above and beyond that which would have been added to the **Omnipool** if the asset balance changes had been due to liquidity events alone.

Thus $Q_i^+ = Q_i + \Delta Q_i + \Delta L$ and we find:

$$p_i^{Q+} = p_i^Q \Leftrightarrow \frac{Q_i + \Delta Q_i + \Delta L}{R_i + (1 - f_R)\Delta R_i} = \frac{Q_i}{R_i} \Rightarrow \frac{Q_i + \Delta Q_i + \Delta L}{R_i + (1 - f_R) R_i \frac{-\Delta Q_i}{Q_i + \Delta Q_i}} = \frac{Q_i}{R_i},$$

which may be simplified to:

$$\Delta L = -\Delta Q_i \left(1 + (1 - f_R)\frac{Q_i}{Q_i + \Delta Q_i}\right). \tag{3.11}$$

Note that $\Delta L < 0$, as it should be in order to 'recover' the originally lower price of asset i before it was removed from the pool. This amount is *not*, however, added to `LRNA` associated with asset i, Q_i. Rather, ΔL is accounted in the `LRNA` imbalance state variable L, and L is later 'rebalanced' during an asset-for-asset swap (cf. Sections 3.3.5 and 3.3.6). Thus, the state transition for L is:

$$L^+ = L + \Delta L = L - \Delta Q_i \left(1 + (1 - f_R)\frac{Q_i}{Q_i + \Delta Q_i}\right);$$

(c) the new quantity of `LRNA` associated with asset i is:

$$Q_i^+ = Q_i + \Delta q^\alpha.$$

2. When asset i is bought from the **Omnipool** at a specified quantity $\Delta r_i^\alpha > 0$:

(a) the quantity $\Delta \hat{R}_i < 0$ returned from the swap mechanism includes the asset fee amount, i.e.

$$(1 - f_R)\Delta \hat{R}_i = -\Delta r_i^\alpha \Rightarrow \Delta \hat{R}_i = -\frac{1}{1 - f_R}\Delta r_i^\alpha;$$

(b) the quantity of `LRNA` $\Delta Q_i > 0$ required for this purchase, including fees, is computed from the swap mechanism as:

$$\Delta Q_i := -Q_i \frac{\Delta \hat{R}_i}{R_i + \Delta \hat{R}_i} > 0; \tag{3.12}$$

which may be expressed in terms of the amount bought, Δr_i^α, as

$$\Delta Q_i = Q_i \frac{\Delta r_i^\alpha}{(1 - f_R) R_i - \Delta r_i^\alpha}.$$

The new quantity of `LRNA` associated with asset i after the swap, then, is:

$$Q_i^+ = Q_i + \Delta Q_i;$$

(c) a *ledger update* is performed to update the value of the `LRNA` imbalance state variable L—note that this has exactly the same form as when a specified quantity of `LRNA` is sold into the **Omnipool**, since this is still a purchase of asset i in exchange for `LRNA`. Thus (cf. equation [3.11])

$$\Delta L = -\Delta Q_i \left(1 + (1 - f_R) \frac{Q_i}{Q_i + \Delta Q_i}\right),$$

and the state transition for L is:

$$L^+ = L + \Delta L = L - \Delta Q_i \left(1 + (1 - f_R) \frac{Q_i}{Q_i + \Delta Q_i}\right);$$

(d) the new quantity of asset i is:

$$R_i^+ = R_i + \Delta R_i = R_i - \Delta r_i^\alpha,$$

since the total change in the **Omnipool** balance of asset i is simply (the negative of) that amount Δr_i^α transferred to the agent, net of fees.

3. When asset i is sold into the **Omnipool**,[9]

(a) the change in the **Omnipool** quantity of asset i is the amount $\Delta R_i > 0$ added by the protocol:

$$R_i^+ = R_i + \Delta R_i;$$

[9] Again, it should be emphasized that as of this writing this is *not* an available mechanism to a trader—the calculations below are a formal expression of the computation required for an asset-for-asset swap as explained further in Section 3.3.4.

(b) the quantity of LRNA returned by the swap, $\Delta Q_i > 0$, is:

$$\Delta Q_i := -Q_i \frac{\Delta R_i}{R_i + \Delta R_i} < 0, Q_i^+ = Q_i + \Delta Q_i;$$

(c) the protocol fee $f_Q \in [0,1)$ is applied to the *ex ante* amount ΔQ_i returned by the swap mechanism. The total fee collected is used to *offset* the LRNA imbalance L in the asset-for-asset swap mechanism—this is described further in Sections 3.3.5 and 3.3.6.

State updates (local)

When LRNA is sold into the **Omnipool**,

(a) the agent's holding of LRNA declines by the amount Δq^α sold:

$$q^{\alpha+} = q^\alpha + \Delta q^\alpha;$$

(b) the agent receives asset i net of fees—when the amount of LRNA is specified then the agent receives:

$$\Delta r_i^\alpha = -(1 - f_R)\Delta R_i,$$

where ΔR_i is given in equation (3.10), while if the amount of asset i is specified then the agent simply receives that amount Δr_i^α. In either case,

$$r_i^{\alpha+} = r_i^\alpha + \Delta r_i^\alpha.$$

3.3.5 Swap asset i with asset j (sell side)

An asset-to-asset swap (cf. Swap.ipynb) is a formal composition of *two* asset-to-LRNA swaps (cf. Section 3.3.4 above). A trader can specify either a *sell* or a *buy* side of the swap. The *sell side* of a swap has a trader specifying an amount of asset i, $\Delta r_i^\alpha < 0$, to sell to the protocol in return for an amount of asset j, $\Delta R_j > 0$. The protocol computes the amount of LRNA that would be returned (as if Δr_i^α were an input to an asset-for-LRNA swap) from the sale of Δr_i^α and uses it to compute (as if it were an input to a LRNA-for-asset swap) the amount ΔR_j.

Requirement

An asset-for-asset swap keeps both Y_i and Y_j invariant during the swap sequence (cf. Section 3.3.4), although these values will generally change when protocol and/or asset fees are non-zero. In addition, absent fees the relation $\Delta Q_i + \Delta Q_j = 0$ is also enforced, as the 'output' quantity of `LRNA` from the first leg, $\Delta Q_i < 0$, is used as the 'input' quantity of `LRNA`, $\Delta Q_j > 0$, for the second leg.

Constraints

1. When asset i is sold into the **Omnipool**, the change in agent's α holdings of the asset is negative:

$$\Delta r_i^\alpha < 0;$$

2. When asset i is sold into the **Omnipool**, the trader must have at least the amount Δr_i^α to trade:

$$r_i^\alpha \geq -\Delta r_i^\alpha.$$

State updates (global)

1. The change in the **Omnipool** quantity of asset i, $\Delta R_i > 0$, equals the amount added by α:

$$\Delta R_i := -\Delta r_i^\alpha > 0, R_i^+ = R_i + \Delta R_i.$$

2. The amount of `LRNA` associated with asset i following the sale changes by $\Delta Q_i < 0$, as given in the asset-to-`LRNA` swap direction (cf. above):

$$\Delta Q_i = -Q_i \frac{\Delta R_i}{R_i + \Delta R_i}, Q_i^+ = Q_i + \Delta Q_i.$$

3. The quantity $-\Delta Q_i$ is assessed the protocol fee f_Q, and the remainder is added as $\Delta Q_j > 0$ to the balance of `LRNA` associated with asset j:

$$\Delta Q_j = -(1 - f_Q)\Delta Q_i, Q_j^+ = Q_j + \Delta Q_j.$$

4. The amount of asset j, $\Delta R_j < 0$, generated by selling ΔQ_j in to the protocol (including fees) is given by the LRNA-to-asset swap direction (cf. above):

$$\Delta R_j = -R_j \frac{\Delta Q_j}{Q_j + \Delta Q_j};$$

the transactions fee f_R is applied to ΔR_j, and the total fee amount is added to the quantity of asset j. The new quantity of asset j after the swap, then, is:

$$R_j^+ = R_j + (1 - f_R)\Delta R_j.$$

5. The total fee, $-\Delta Q_i f_Q > 0$, is used to offset the imbalance of LRNA recorded in the ledger L. Recall (cf. Section 3.3.4 and equation [3.11]) that this (negative) balance accrues whenever a LRNA-for-asset swap is performed, and indicates the departure (in terms of LRNA) of asset prices from their pre-swap values. It is to this balance that the protocol fees $-\Delta Q_i f_Q$ are applied.

   Formally, this fee amount is burned and is credited against the LRNA imbalance ledger L (cf. Section 4.1.5 for further details regarding this mechanism). If $-L \geq -\Delta Q_i f_Q$, then the ledger balance exceeded the total fee collected, and so the change in the ledger balance is just the total amount credited:

$$\Delta L = -\Delta Q_i f_Q.$$

   If, however, $-\Delta Q_i f_Q > -L$, then the total fee collected was enough to completely 'balance the books' for the LRNA imbalance. In this case

$$\Delta L = -L,$$

   and the excess amount $L - \Delta Q_i f_Q > 0$ is used to increase the amount of LRNA Q_{HDX} associated with the **Omnipool** governance asset HDX:

$$\Delta Q_{HDX} = L - \Delta Q_i f_Q.$$

We may condense the above to:

$$\Delta L = \min\{-\Delta Q_i f_Q, -L\};\ L^+ = L + \Delta L, \tag{3.13}$$

$$\Delta Q_{HDX} = -(\Delta Q_i f_Q + \Delta L);\ Q^+_{HDX} = Q_{HDX} + \Delta Q_{HDX}. \tag{3.14}$$

State updates (local) When asset i is sold into the **Omnipool**,

1. the agent's holding of asset i declines by the amount $\Delta r_i^\alpha < 0$ sold:

$$r_i^{\alpha+} = r_i^\alpha + \Delta r_i^\alpha;$$

2. the agent receives asset j net of fees in the amount $\Delta r_j^\alpha > 0$:

$$\Delta r_j^\alpha = -(1 - f_R)\Delta R_j, r_j^{\alpha+} = r_j^\alpha + \Delta r_j^\alpha.$$

3.3.6 Swap asset i with asset j (buy side)

An asset-to-asset swap (cf. Swap.ipynb) is a composition of *two* asset-to-LRNA swaps (cf. Section 3.3.4 above). A trader can specify either a *sell* or a *buy* side of the swap. The *buy side* of a swap has a trader specifying an amount of asset j, $-\Delta R_j > 0$, to buy from the protocol in return for an amount of asset i provided, $\Delta r_i^\alpha < 0$. The protocol computes the amount of LRNA that would be required to purchase ΔR_j from the pool (as if ΔR_j were an output of a LRNA-for-asset swap, net of fees) and uses that LRNA amount to compute (as if it were an output of an asset-for-LRNA swap) the amount of asset i required to be sold, Δr_i^α.

Constraints

1. When asset j is bought from the **Omnipool**, the change in agent's α holdings of the asset is negative:

$$\Delta r_i^\alpha < 0;$$

2. When asset j is bought from the **Omnipool**, the trader must have at least the amount Δr_i^α to trade:

$$r_i^\alpha \geq -\Delta r_i^\alpha.$$

State updates (global)

1. The change in the amount of `LRNA` associated with asset j, $\Delta Q_j > 0$, is the amount that would be required to generate (from the `LRNA`-to-asset swap) that amount $\Delta\tilde{R}_j < 0$ which, when fees are deducted, corresponds to the requested amount ΔR_j. To arrive at this, note that for the latter to be true, $(1-f_R)\Delta\tilde{R}_j = \Delta R_j$. This implies:

$$\Delta Q_j = -Q_j \frac{\Delta\tilde{R}_j}{R_j + \Delta\tilde{R}_j} = -Q_j \frac{\frac{1}{1-f_R}\Delta R_j}{R_j + \frac{1}{1-f_R}\Delta R_j} = -Q_j \frac{\Delta R_j}{(1-f_R)R_j + \Delta R_j}. \tag{3.15}$$

2. The amount of `LRNA` associated with asset i is that amount $\Delta Q_i < 0$ corresponding to the amount of `LRNA` $\Delta Q_j > 0$ after the protocol fee would have been removed from ΔQ_i:

$$(1-f_Q)\Delta Q_i = -\Delta Q_j \Rightarrow \Delta Q_i = -\frac{1}{1-f_Q}\Delta Q_j, Q_i^+ = Q_i + \Delta Q_i.$$

3. The amount of asset i, $\Delta R_i > 0$, that would have generated the amount ΔQ_i from the asset-to-`LRNA` swap is given by:

$$\Delta R_i = -R_i \frac{\Delta Q_i}{Q_i + \Delta Q_i}, R_i^+ = R_i + \Delta R_i.$$

4. The total fee, $-\Delta Q_i f_Q > 0$, is used to offset the imbalance of `LRNA` recorded in the ledger L. This proceeds as in the previous section (cf. Section 3.3.5), resulting in:

$$\Delta L = \min\{-\Delta Q_i f_Q, -L\};\ L^+ = L + \Delta L, \tag{3.16}$$

$$\Delta Q_{HDX} = -(\Delta Q_i f_Q + \Delta L);\ Q_{HDX}^+ = Q_{HDX} + \Delta Q_{HDX}. \tag{3.17}$$

5. The final amount of the reserve asset j, R_j^+, is given by the net change of transferring $-\Delta R_j$ to the agent—note that this already accounts for fees being added back into the reserve (since $-\Delta R_j$ is the *net transfer* to agent α):

$$R_j^+ = R_j + \Delta R_j.$$

State updates (local) When asset j is bought from the **Omnipool**,

1. the agent buys asset j net of fees in the amount $\Delta r_j^\alpha > 0$:

$$\Delta r_j^\alpha = -\Delta R_j, r_j^{\alpha+} = r_j^\alpha + \Delta r_j^\alpha;$$

2. the agent's holding of asset i declines by the amount $\Delta r_i^\alpha < 0$ needed to pay for asset j:

$$\Delta r_i^\alpha = -\Delta R_i, r_i^{\alpha+} = r_i^\alpha + \Delta r_i^\alpha.$$

4 Economic Implications

4.1 Pool Viability

This Report does not examine the medium-to-long term horizon of the **Omnipool**'s 'tokenomics', i.e. the conclusions that can be drawn from particular assumptions regarding e.g. trading behavior (such as arbitrage) and the composition of **Omnipool** participants (such as the relative mix of liquidity providers and traders). However, some foundational conclusions regarding the viability of the **Omnipool** independent of behavior may be drawn (cf. Sections 4.1.1 - 4.1.4), while a brief analysis of the `LRNA` imbalance ledger under a minimal `HDX` valuation assumption is provided in Section 4.1.5.

4.1.1 Exponential Withdrawal

As with any Constant Function Market Maker (CFMM) framework utilizing an invariant of the form $Y_i = R_i Q_i$ for an asset i, it requires an exponentially increasing quantity of `LRNA` to be contributed to the pool in order to withdraw, in the limit only, the entire asset i balance from the **Omnipool**. Whether or not such activity would transpire in response to a change in external market conditions, such as an increase in the external market price of asset i, is addressed in the following section.

4.1.2 Finite Price Convergence

Following an increase in an external price of an asset contained in the **Omnipool**, trading activity that acts to increase the **Omnipool** price of that asset does not exhaust the pool of the asset. In other words, for any price 'wedge' $\Delta p := p_i^e - p_i > 0$, where p_i^e is a reference price of asset i in an external market, and p_i is the initial price of that asset in the **Omnipool**, trading activity starting from asset i's initial balance R_i that results in the convergence of $p_i \to p_i^e$ causes an outflow of asset i from the pool, $\Delta R_i < 0$, such that

$$R_i > -\Delta R_i.$$

We briefly demonstrate this for the `LRNA`-for-asset mechanism (a similar analysis may be performed for the composition of two swap mechanisms, to achieve an asset-for-asset trade, where one or both of the protocol and asset fees are non-zero).

We suppose without loss of generality that prices are denominated in `LRNA`. Then there is a sequence of K trades $\Delta Q_i^1, \ldots, \Delta Q_i^K$ such that at the conclusion of these trades,

$$p_i' = \frac{Q_i'}{R_i'} = p_i^e,$$

where p_i' is the **Omnipool** price after the sequence has concluded (the 'final' price), R_i' is the final **Omnipool** balance of asset i, and Q_i' is its associated balance of `LRNA`.

We know that following any swap in of `LRNA` ΔQ_i^k,

$$p_i^{k+1} = \frac{Q_i^k + \Delta Q_i^k}{R_i^k + \Delta R_i^k},$$

where R_i^k and Q_i^k are pool balances of asset i and `LRNA` immediately before the swap, respectively, and p_i^{k+1} is the new price after the swap. From equation (3.10) it is straightforward to show that this is equivalent to:

$$p_i^{k+1} = \frac{(Q_i^k + \Delta Q_i^k)^2}{R_i^k \left(Q_i^k + f_R \Delta Q_i^k\right)},$$

and this may be simplified further by introducing the rate of change of the pool balance of `LRNA` after the k-th swap, $q_i^k := \Delta Q_i^k / Q_i^k > 0$, and using the fact that $p_i^k = Q_i^k / R_i^k$:

$$p_i^{k+1} = p_i^k \frac{\left(1 + q_i^k\right)^2}{1 + f_R q_i^k}.$$

This difference equation for prices allows us to immediately write the relationship between the final price $p_i' = p_i^e$ and the initial price p_i, for a sequence of swaps with associated rates of change $\{q_i^k\}_{k=1}^K$:

$$p_i^e = p_i' = p_i \prod_{k=1}^{K} \frac{\left(1 + q_i^k\right)^2}{1 + f_R q_i^k}. \tag{4.1}$$

One may also show that, after the K trades are performed, the final **Omnipool** balance R_i' of asset i after starting from an initial balance R_i is

$$R_i' = R_i \prod_{k=1}^{K} \frac{1 + f_R q_i^k}{1 + q_i^k}$$

(recall that the presence of f_R is due to the fact that for every trade ΔR_i^k, a fee amount $f_R \Delta R_i^k$ is returned to the pool following the swap). Thus, for the above claim to be true it must be the case that

p_i^e is reached after a *finite* number of trades, for in that case

$$R_i' = R_i \prod_{k=1}^{K} \frac{1 + f_R q_i^k}{1 + q_i^k} > 0$$

whenever $R_i > 0$, i.e. the reserve asset balance has not been reduced to zero.

Suppose to the contrary that $K = \infty$, i.e. it takes an infinite number of trades to reach p_i^e (this would correspond to the circumstance under which the reserve asset i was, in the process, completely removed from the **Omnipool** due to trading). Then from equation (4.1) we have

$$\frac{p_i^e}{p_i} = \lim_{K \to \infty} \prod_{k=1}^{K} \frac{\left(1 + q_i^k\right)^2}{1 + f_R q_i^k} = \infty,$$

since for any k, $(1 + q_i^k) > (1 + f_R q_i^k) > 1$. With both the external market price p_i^e and the initial price p_i finite, this results in a contradiction. Hence $K < \infty$ and consequently the final reserve balance R_i' is non-zero, and the external market price is achieved from a finite number of `LRNA`-for-asset swaps.

4.1.3 Reciprocal Swaps

A related analysis to Section 4.1.2 involves whether or not an entity can remove value from the **Omnipool** by trading, say, an asset i into the pool, receiving `LRNA`, and then immediately trading `LRNA` in to receive asset i once again.[10] If the final net value of asset i received is greater than the initial value traded in, then the trader has earned a riskless positive return at the expense of the pool.

The **Omnipool** is not vulnerable to value extraction from reciprocal swaps between an asset i and `LRNA`. From Section 3.3.5, we may perform the asset-to-asset procedure with $j = i$, which replicates the reciprocal swap process described above with an initial amount of asset i sold to the pool. From that analysis, an entity first sells $\Delta R_i > 0$ to the pool in return for `LRNA`, which is assessed the protocol fee f_Q. The net amount is then sold again into the pool, and the entity receives asset i back after the asset fee f_R has been assessed. Denoting the final amount of asset i received by $\Delta R_i' > 0$,[11] and introducing the rate of change of the **Omnipool**'s initial balance R_i of asset i from the entity's initial

[10] Although as of this writing the asset-for-`LRNA` mechanism is inaccessible to a trader, its potential future accessibility allows us to conduct this "what if" viability analysis.

[11] For notational convenience we depart here from convention and express this as a positive amount.

asset sale ΔR_i by $r_i := \Delta R_i / R_i$, the relationship between $\Delta R'_i$ and ΔR_i may be expressed as

$$\Delta R'_i = \Delta R_i (1 - f_R) \left(\frac{1 - f_Q}{1 - f_Q \frac{r}{1+r}} \right).$$

Since $(1 - f_R) < 1$ whenever $f_R \neq 0$, and $(1 - f_Q)/(1 - f_Q(r/(1 + r))) < 1$ whenever $f_Q \neq 0$, the above relation implies that $\Delta R'_i < \Delta R_i$ always, i.e. the entity always receives less of asset i when performing the reciprocal swap. (As an aside, we note that if one of f_R or f_Q is zero the inequality $\Delta R'_i < \Delta R_i$ still holds, while if $f_R = f_Q = 0$ the entity receives exactly what they put into the pool initially, i.e. $\Delta R'_i = \Delta R_i$, and the state of the pool after the reciprocal swaps are performed is the same as its state before the swaps.)

This analysis is sufficient to claim that the value of the reserve asset i received from the reciprocal swaps is less than the value of the asset initially sold into the pool: because more of asset i is added to the **Omnipool**, while the associated balance of `LRNA` declines (because of the protocol fee), the price of asset i in terms of `LRNA` also declines. This means that the post-reciprocal-swap value of asset i received by the entity is less than the value of the amount of the asset initially sold to the pool at the pre-reciprocal-swap price. Finally, in the plausible case where there exists a price in a reference asset, such as `USD`, which is insensitive to changes in the price of asset i derived from the **Omnipool**, the lower balance of asset i received means it is not possible to profit from reciprocal swaps on the external market.

4.1.4 Discretionary Pool Shares

The protocol accumulates pool shares of a non-`LRNA` asset whenever the price of an asset at liquidity redemption is below the asset price at the time of liquidity provision (cf. Section 3.3.1). This accumulation compensates the protocol for assuming part of the risk of a single-asset liquidity position, and adds discretionary power to future mechanisms where such pool share balances are used for ancillary services.

4.1.5 `HDX` Support

As discussed in Section 3.3, the `LRNA` imbalance ledger L accounts for deviations in the amount of `LRNA` in the **Omnipool** that may be ascribed to its market-making activity, and to its taking part in risk-sharing with liquidity providers. Although the primary aim of the **Omnipool** is to facilitate asset exchange, its *means* of doing so—namely, the creation of a pool of assets available for exchange—is facilitated by attempting to keep the pool's position in LRNA that amount which would have been obtained had the **Omnipool**'s asset balances been created solely through liquidity events. This is the reason behind crediting burned protocol fees to L—these fees are *actually removed from the economy* and hence help to redress the imbalance that is being kept track of by the ledger L.

Of course, there is no guarantee that fees are sufficient to exactly cover L, and so in low trading volume environments this fee burning and crediting mechanism fulfills this offsetting characteristic. But it is also worth examining the important case where fees are *more than enough* to cover L. This indicates the situation where the **Omnipool**'s trading volume is high enough that its aggregate LRNA balance can be viewed as if it was derived from liquidity events. The question then arises: should fees in excess of the ledger imbalance, i.e. that amount remaining after crediting results in $L = 0$, be burnt? This is potentially a non-trivial issue since allowing the imbalance to move in the other direction, i.e. to have relatively less LRNA in the **Omnipool** from burning after the imbalance ledger L reaches zero, may exacerbate rather than mitigate impermanent loss.

Rather than burning this excess, the mechanism adopted by the **Omnipool** is to utilize the excess to increase the value of the governance token of the ecosystem, HDX. This is accomplished by increasing the quantity of LRNA associated with HDX, Q_{HDX}, by the excess fee amount. This has the effect of increasing the price of HDX in terms of LRNA.[12] This mechanism is appealing because (under a minimal behavioral assumption on stakeholder preferences[13]) it provides a positive correlation between participating in the ecosystem via the HDX token and the performance of that ecosystem as measured by protocol fees generated from a high trading volume.[14]

An illustrative calculation reveals the increase in the value of the HDX token in the **Omnipool** that obtains for a particular trading volume, when that volume allows a ledger imbalance L to be fully credited with an excess amount available to be placed in the LRNA balance associated with HDX. To that end, consider initially a single asset-for-asset swap, where specified amount $\Delta R_i > 0$ of a non-HDX asset i is sold into the **Omnipool** in return for another non-HDX asset. Following the mechanism given in Section 3.3.5, this yields an amount $\Delta Q_i < 0$ that is removed from the LRNA associated with asset i, and is assessed the protocol fee f_Q before the residual is used as an input into the LRNA-for-

[12] It is uncertain if this would increase the price of HDX in terms of an external reference unit of account, such as USD, since the excess quantity of LRNA may drive down its price in that unit of account.

[13] Namely, that participation valuation is at least partly measured by the value of a stakeholder's HDX token holding.

[14] It is worth recalling here that this is independent of the return to liquidity providers from an increase in *asset* fees, which are returned to the asset pool.

asset leg.

Suppose that earlier fees have reduced the LRNA imbalance to zero, so that the collected fee from the trade volume ΔR_i, $-f_Q \Delta Q_i$, is added to the LRNA associated with HDX in the **Omnipool**, i.e. $Q^+_{HDX} = Q_{HDX} - f_Q \Delta Q_i$. Since $p^Q_{HDX} = Q_{HDX}/R_{HDX}$, the addition of the fee raises the price of HDX to:

$$p^{Q+}_{HDX} = \frac{Q^+_{HDX}}{R^+_{HDX}} = \frac{Q_{HDX} - f_Q \Delta Q_i}{R_{HDX}}$$

(note that $R^+_{HDX} = R_{HDX}$ since the amount of HDX in the **Omnipool** is unchanged). From the swap mechanism we know

$$\Delta Q_i = -Q_i \frac{\Delta R_i}{R_i + \Delta R_i},$$

so that

$$p^{Q+}_{HDX} = \frac{Q_{HDX}}{R_{HDX}} + \frac{f_Q Q_i \frac{\Delta R_i}{R_i + \Delta R_i}}{R_{HDX}},$$

or

$$p^{Q+}_{HDX} = p^Q_{HDX} + f_Q Q_i \left(1 + \frac{\Delta R_i}{R_i}\right)^{-1}.$$

From this relation we can derive the change in the LRNA price of HDX from an initial value p to a final value p' (dropping the 'Q' superscript and HDX subscript for notational convenience), for a sequence of trades among and between the N non-HDX assets in the **Omnipool**, when the amount of HDX in the **Omnipool** is unchanged at a value R. For concreteness assume that for each asset $n = 1, \ldots N$ there are precisely $K(n)$ trades where asset n is sold into the **Omnipool**. Then the final price p' may be expressed as:

$$p' = p + \frac{f_Q}{R} \sum_{n=1}^{N} \sum_{k=1}^{K(n)} Q^k_n \left(1 + \frac{\Delta R^k_n}{R_k}\right)^{-1},$$

where R^k_n and Q^k_n are the amounts of asset n and LRNA associated with asset n immediately prior to the k-th transaction involving asset n, respectively, and ΔR^k_n is the amount of asset n sold into the **Omnipool** for that k-th transaction. Letting $r^k_n := \Delta R^k_n / R^k_n$ denote the rate (relative to its balance in the **Omnipool**) at which asset n is deposited for that k-th transaction, the above can also be written

$$p' = p + \frac{f_Q}{R} \sum_{n=1}^{N} \sum_{k=1}^{K(n)} Q^k_n \left(\frac{r^k_n}{1 + r^k_n}\right).$$

Examining the change in price of HDX in terms of LRNA reveals that this mechanism supports:

1. an appreciation of the value of HDX as the protocol fee f_Q is increased;

2. an appreciation of the value of HDX as the asset deposit rate r_n^k is increased.

4.2 Impermanent Loss

Impermanent Loss (IL) is a measurement of the opportunity cost of providing liquidity to a pool of assets—while the value of those assets may fluctuate on markets external to (and potentially uncorrelated with) the pool, asset value *within* the pool is constrained to their relative quantities. This means that external market movements can induce changes in the relative quantities of pool assets that, while equilibrating external and internal measures of value (prices), results in value lost to the liquidity provider. The following provides an overview of IL within the **Omnipool** for non-HDX assets,[15] for the case without (Section 4.2.1) and with (Section 4.2.2) transactions fees.

4.2.1 IL without transactions fees

The current pool specification is designed to enable liquidity providers (LPs) to transparently understand their IL when providing liquidity. To facilitate this understanding, consider first liquidity provision in a single asset to a pool without associated trading (swap) fees. As described in Section 3.3.1, an LP α adding reserve to an existing asset i contributes an amount $\Delta r_i^\alpha = \Delta R_{i0} < 0$ at the time $t = 0$ of contribution. The LP receives pool shares $\Delta s_i^\alpha > 0$ in the amount

$$\Delta s_i^\alpha := -\frac{S_{i0}}{R_{i0}} \Delta R_{i0} > 0.$$

Recall from Section 3.3.1 that S_{i0} is the 'initial condition' of outstanding pool shares for the existing asset i in the pool at the time of contribution. The protocol records the price of the asset at the time of contribution for LP α, p_i^α, which is the ratio Q_{i0}/R_{i0}.

As described in Section 3.3.3, the LP is entitled to a particular value V_i^α when they decide to redeem their pool shares Δs_i^α, where now $\Delta s_i^\alpha < 0$ to reflect the transfer of pool shares from the agent to the protocol. In order to guarantee V_i^α, the protocol either provides *both* a token reserve amount, Δr_i^α, and an amount of LRNA, Δq_i^α, if the price of asset i has risen since contribution, or collects a fraction of the shares redeemed and provides only a token reserve amount Δr_i^α if the price

[15] Although a comprehensive IL analaysis for HDX is not included in this Report, it is speculated that HDX IL is bounded from above by the IL associated with non-HDX assets.

of asset i has fallen since contribution. In either case (cf. equation 3.7),

$$V_i^\alpha = p_i^Q \Delta r_i^\alpha + \Delta q_i^\alpha = -p_i^Q \left(\frac{2p_i^Q}{p_i^Q + p_i^\alpha} \frac{\Delta s_i^\alpha}{S_i} R_i \right) = -\frac{R_i}{S_i} \frac{2(p_i^Q)^2}{p_i^Q + p_i^\alpha} \Delta s_i^\alpha > 0.$$

This is the value that an LP expects to receive when converting their pool shares back into asset i, and is usually called the "pool value" V_P of their liquidity provision when computing IL. This may be compared to the value of their original liquidity provision, ΔR_{i0}, if it had instead remained outside of the pool–in terms of LRNA this is:

$$V_H := p_i^Q \Delta R_{i0},$$

where V_H is interpreted as the "hold value" of the reserve contribution held outside of the pool. Note that here $\Delta R_{i0} > 0$ since it remains in the hands of the LP and is not contributed to the pool.

The opportunity cost of liquidity provision, then, is the difference between V_P and V_H. This is usually expressed as a percentage change in the hold value of the asset, of the value of providing the asset to the pool, and is the **impermanent loss** (IL):

$$IL := \frac{V_P - V_H}{V_H} = \frac{V_P}{V_H} - 1.$$

Substituting in the above, this yields

$$IL = -\frac{1}{p_i^Q \Delta R_{i0}} \frac{R_i}{S_i} \frac{2(p_i^Q)^2}{p_i^Q + p_i^\alpha} \Delta s_i^\alpha - 1 = \frac{2p_i^Q}{p_i^Q + p_i^\alpha} \frac{R_i}{S_i} \frac{S_{i0}}{R_{i0}} - 1,$$

or

$$IL = \frac{2p_i^Q}{p_i^Q + p_i^\alpha} \frac{R_i/S_i}{R_{i0}/S_{i0}} - 1. \tag{4.2}$$

To simplify this, note that the relative growth rates of the reserve asset balance R_i and its pool share amount S_i must be compared, since

$$\frac{R_i/S_i}{R_{i0}/S_{i0}} = \frac{R_i/R_{i0}}{S_i/S_{i0}}.$$

When a liquidity event occurs (adding liquidity to an existing asset in the pool, adding a new asset to the pool, or withdrawing liquidity from the pool) the aforementioned mechanisms preserve R_i/S_i and the asset price remains unchanged. But when a trade event occurs (swapping an asset for LRNA, or swapping LRNA for an asset) the amount of a risk asset and its associated price changes, while pool shares remain unchanged.

From the invariant quantity R_iQ_i, we know that following a swap event, which (say) changes the amount of reserve from R_i to R_i' and the price (in terms of LRNA) from p_i to p_i', we have

$$R_i = \left(\frac{p_i'}{p_i}\right)^{1/2} R_i',$$

meaning that for *any* sequence of swap and liquidity events, commencing from a pool state $(R_{i0}, S_{i0}, p_i^\alpha)$ and arriving at a pool state (R_i, S_i, p_i^Q),

$$R_{i0} = \left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2} \frac{S_{i0}}{S_i} R_i,$$

meaning

$$\frac{R_i/S_i}{R_{i0}/S_{i0}} = \left(\frac{p_i^Q}{p_i^\alpha}\right)^{-1/2}.$$

Substitution of this relation into (4.2) finally yields

$$IL = \frac{2\sqrt{p_i^Q p^\alpha}}{p_i^Q + p^\alpha} - 1.$$

This measure of IL is identical to that experienced by a pool comprised of two assets behaving according to a simple constant product market maker (CPMM) "$x \cdot y = k$" relation in the absence of fees, such as Uniswap 2.0. In other words, under the conditions where the LP must receive part of their pool share value in LRNA (and not exclusively in token), their impermanent loss is the loss that would be obtained *as if* they had provided both the asset i and LRNA to a Uniswap 2.0 pool.

As an alternative characterization, and proceeding directly from the LP's received value (equation 3.4), we have

$$V_P = V_i^\alpha = \frac{2p_i^Q p_i^\alpha}{p_i^Q + p_i^\alpha}\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2} \Delta R_{i0}.$$

Coupled with

$$V_H := p_i^Q \Delta R_{i0}$$

we conclude that

$$IL := \frac{V_P}{V_H} - 1 = \frac{\frac{2p_i^Q p_i^\alpha}{p_i^Q + p_i^\alpha}}{p_i^Q}\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2} - 1 = \frac{2\sqrt{p_i^Q p^\alpha}}{p_i^Q + p^\alpha} - 1. \tag{4.3}$$

The above characterization also makes clear that the IL burden on the LP does not depend upon the sign change in the price of the asset. If the asset price falls from the time of liquidity provision to the time of withdrawal, the LP does not receive a combination of asset i and LRNA—rather, they receive all of their value in asset i. But from (4.3), the LP again experiences IL as if they had provided amounts of asset i and LRNA to a Uniswap 2.0 pool. This is because in this case part of the value of the LP's pool shares is transferred to the protocol (i.e. $\Delta B_i > 0$), as part of the risk-sharing arrangement between the protocol and the LP.

4.2.2 IL mitigation from fees

The **Omnipool** protocol will apply a transaction fee whenever an asset is swapped out of the pool. This transactions fee, f_R, when applied to the asset is denominated in the asset and is *returned* to the pool, with the remainder of the asset provided to the trader. Thus, for every transaction in which an asset is swapped out of the pool, an LP with exposure in that asset will command a larger amount of the asset (via their held pool shares) than they would otherwise obtain if fees were absent.

To derive the IL with fees included, return to equation (3.4), which derives the value the LP is entitled to when withdrawing their pool shares $\Delta s_i^\alpha < 0$:

$$V_i^\alpha = \frac{2p_i^Q p_i^\alpha}{p_i^Q + p_i^\alpha} \left(\frac{p_i^Q}{p_i^\alpha} \right)^{1/2} \Delta R_{i0}.$$

When fees are included, the LP's pool shares command a reserve amount

$$\Delta R_i^\alpha = -\frac{\Delta s_i^\alpha}{S_i} R_i' = -\frac{\Delta s_i^\alpha}{S_i}(R_i + F_i),$$

where we denote by R_i' the amount of asset i in the pool, and by R_i the amount that would have been in the pool net of the fees actually added, F_i. Recall that ΔR_i^α is the amount in the asset i pool attributed to the original liquidity contribution ΔR_{i0}, i.e.

$$\Delta R_{i0} = \left(\frac{p_i^Q}{p_i^\alpha} \right)^{1/2} \Delta R_i^\alpha.$$

Substitution from the above into the LP's value entitlement thus yields:

$$V_i^\alpha = \frac{2p_i^Q p_i^\alpha}{p_i^Q + p_i^\alpha}\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2} \Delta R_{i0} = \frac{2p_i^Q p_i^\alpha}{p_i^Q + p_i^\alpha}\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2}\left(\frac{p_i^Q}{p_i^\alpha}\right)^{1/2} \Delta R_i^\alpha = \tag{4.4}$$

$$-\frac{2(p_i^Q)^2}{p_i^Q + p_i^\alpha}\frac{\Delta s_i^\alpha}{S_i}(R_i + F_i)\,. \tag{4.5}$$

Recall that V_i^α is V_P, the value of the LP's contribution in the pool (now including fees), and $V_H = p_i^Q \Delta R_{i0}$ is the value of the original contribution held outside of the pool, at the time of withdrawal. Thus, the IL for the LP when fees are included, denoted $IL(f_R)$ for the transactions fee percentage f_R, is:

$$IL(f_R) = IL - \frac{2p_i^Q}{p_i^Q + p_i^\alpha}\frac{\Delta s_i^\alpha}{S_i}\frac{F_i}{\Delta R_{i0}},$$

indicating that fees mitigate the amount of impermanent loss without fees (IL) by

$$\Delta IL := \frac{2p_i^Q}{p_i^Q + p_i^\alpha}\frac{\Delta F_i^\alpha}{\Delta R_{i0}} > 0,$$

where the share of fees the LP earns is denoted $\Delta F_i^\alpha := -\frac{\Delta s_i^\alpha}{S_i} F_i > 0$. We define *impermanent loss mitigation from fees* to be the quantity ΔIL.

Empirically, ΔIL can be derived by subtracting the *calculated* Uniswap 2.0 impermanent loss for the pool state without fees from the *measured* impermanent loss from the **Omnipool** with a particular fee f_R.

Connecting the transactions fee rate f_R to the impermanent loss mitigation requires following the dependencies of both the fee amount F_i and the price p_i^Q. The fee amount is straightforward: given a trading (swap) volume $\Delta T_i < 0$ on asset i outgoing (i.e. swapped out when `LRNA` is exchanged directly, or swapped out when an asset j is exchanged), the fee amount is just

$$F_i = -f_R \Delta T_i,$$

indicating a linear gain in IL mitigation in f_R when considering this term alone.

4.3 Slippage

A trader interacting with the **Omnipool** can condition their trade upon the state of the **Omnipool** using a *spot price* of an asset j in terms of `LRNA` or an other asset i. In either case, the spot price represents the instantaneous rate of exchange between assets and is dependent only upon the existing quantities of assets in the **Omnipool**. Formally, a spot price p_j^i of an asset j in terms of an asset i is given by the ratio of an infinitesimal amount of asset i provided to the pool, dR_i, to an infinitesimal amount of the desired asset j received from the pool, dR_j:

$$p_j^i = \frac{dR_i}{dR_j}.$$

For exchanges between an asset j and `LRNA`, the notation dQ_j will be used, e.g.

$$p_j^Q = \frac{dQ_j}{dR_j}.$$

Recall that whenever a swap takes place the *invariant* $V_j := R_j Q_j$ is conserved (absent fees) for each leg of an asset-for-`LRNA` exchange. The spot price of an asset in terms of `LRNA`. then, may be derived from the invariant as follows:

$$dV_i = 0 \Rightarrow dR_j Q_j + R_j dQ_j \equiv 0 \Rightarrow \frac{dQ_j}{dR_j} = -\frac{Q_j}{R_j},$$

as is well-known from other CFMM invariant functions.

Deriving the price of an asset j in terms of an asset i, then, requires the spot price of the composed swap mechanisms exchanging asset i for `LRNA`, and `LRNA` for asset j:

$$p_j^i = \frac{dR_i}{dR_j} = \frac{dQ_i}{dQ_i}\frac{dR_i}{dR_j} = \frac{dQ_j/dR_j}{dQ_i/dR_i} = \frac{p_j^Q}{p_i^Q} = \frac{Q_j}{Q_i}\frac{R_i}{R_j},$$

where the mechanism composition has, in the absence of fees, required $dQ_i = dQ_j$ as the amount of `LRNA` bought with asset i is immediately used to buy asset j.

Consider now a trader interested in exchanging asset i for asset j. The profitability of the exchange depends upon the actual *realized price* of the transaction, which in turn depends not only upon the state of the **Omnipool**, but upon the size(s) of the asset quantities traded. This realized price is the actual exchange of an asset i quantity ΔR_i for asset j quantity ΔR_j:

$$\pi_j^i = \frac{\Delta R_i}{\Delta R_j}.$$

Comparing π_j^i and p_j^i indicates the degree to which the state of the **Omnipool**, as summarized by its spot prices, reflects its ability to efficiently predict the price realized by traders when engaging with the **Omnipool** to swap assets. **Slippage** is defined as the degree to which the realized price deviates from the spot price, reflecting an AMM's trade efficiency:

$$SL(i,j) := \frac{\pi_j^i}{p_j^i} - 1.$$

If an AMM is fully efficient, then $\pi_i^j = p_j^i$ and slippage is zero, reflecting the idealization where the spot price fully captures the realized price—such would be the case, for example, in an AMM where the liquidity pools were so large that prices did not move from any trading activity. Otherwise (and more realistically) nearly every AMM implementation experiences, just as in non-AMM markets, some degree of slippage.[16]

4.3.1 Slippage without fees

It is straightforward, using the above definitions, to derive the slippage for the **Omnipool** in the case without fees. As a simple example, consider the case of trading a quantity of `LRNA`, ΔQ_j, in return for a quantity of asset j, ΔR_j from the pool. Using the swap mechanism described in Section 3.3.4 and the above definitions, we have:

$$\pi_j^Q = \frac{\Delta Q_j}{\Delta R_j} = \frac{\Delta Q_j}{R_j \frac{\Delta Q_j}{Q_j + \Delta Q_j}} = \frac{Q_j + \Delta Q_j}{R_j}.$$

From this, the slippage associated with a `LRNA` swap size of ΔQ_j is:

$$SL(Q,j) := \frac{\pi_j^Q}{p_j^Q} - 1 = \left(\frac{Q_j + \Delta Q_j}{R_j} \right) \frac{R_j}{Q_j} - 1 = \frac{\Delta Q_j}{Q_j},$$

[16] For completeness it may be noted that there exist AMM implementations such as *constant sum* market makers, which provide zero slippage at the expense of allowing liquidity to be completely removed from the pool following a price imbalance (cf. Section 4.1.2).

where the second equality follows from the definition of p_j^Q as the spot price of asset j in terms of LRNA. This makes it clear for this simple example that only the relative size of the desired transaction to the reserve quantity contained in the **Omnipool** matters for determining slippage.[17]

A more complicated example is the swap between an asset i in quantity ΔR_i and an asset j in quantity ΔR_j, which is the composition of two swaps, asset-for-LRNA and LRNA-for-asset. Proceeding from the swap mechanisms defined in Section 3.3.4, we have

$$\pi_j^i = \frac{\Delta R_i}{\Delta R_j} = \frac{\Delta R_i}{\frac{R_j Q_i \Delta R_i}{R_i Q_j + (Q_j + Q_i)\Delta R_i}} = \frac{R_i}{R_j}\frac{Q_j}{Q_i} + \left(1 + \frac{Q_j}{Q_i}\right)\frac{\Delta R_i}{R_j}.$$

This realized price implies that the slippage associated with swapping ΔR_i for ΔR_j is:

$$SL(i,j) =:= \frac{\pi_j^i}{p_j^i} - 1 = \left(\frac{R_i}{R_j}\frac{Q_j}{Q_i} + \left(1 + \frac{Q_j}{Q_i}\right)\frac{\Delta R_i}{R_j}\right)\left(\frac{Q_i}{Q_j}\frac{R_j}{R_i}\right) - 1 = \left(1 + \frac{Q_i}{Q_j}\right)\frac{\Delta R_i}{R_i}.$$

Again, in the limit where ΔR_i is very small compared to R_i the slippage approaches zero—but slippage is now exacerbated by the relative quantities of LRNA associated with assets i and j, respectively. This is a direct consequence of the realized price's dependence upon the same relative quantities—as indicated, if the value of asset j is significantly higher than the value of asset i in LRNA, the realized price of asset j in terms of asset i will be high–this is the effect of the ratio Q_j/Q_i in the expression for pi_j^i. Since slippage indicates the deviation from the spot price p_j^i, which offsets this value imbalance by the ratio of asset reserves R_i/R_j (which, it should be noted, would be the spot price for a CFMM with invariant $V = R_i R_j$ and direct swaps between assets, such as would occur in a Uniswap 2.0 asset i-to-asset j pool), this deviation will be larger the less valuable asset j is relative to asset i. This latter quantity is Q_i/Q_j.

4.3.2 Slippage with fees

As discussed, there are two fees in the **Omnipool** specification: a protocol fee f_Q assessed on outbound amounts of LRNA, and an asset fee f_R assessed on outbound amounts of an asset. For swaps between an asset j and LRNA, a fee acts as a slippage multiplier—for example, in the simple case where a quantity ΔQ_j LRNA is swapped for an asset j in quantity ΔR_j, we have

$$\pi_j^Q(f_R) = \frac{\Delta Q_j}{(1 - f_R)\Delta R_j} = \frac{1}{1 - f_R}\pi_j^Q,$$

[17] It is straightforward to show the same conclusion holds when trading asset i in return for LRNA.

where π_j^Q is the realized price without fees. Then slippage is just:

$$SL(Q, j, f_R) = \frac{\pi_j^Q(f_R)}{p_j^Q} - 1 = \frac{f_R}{1 - f_R} + \frac{1}{1 - f_R}\frac{\Delta Q_j}{Q_j}.$$

Thus, the asset fee f_R has a nonlinear effect on slippage, although for low fees the overall effect is small.

For swaps between an asset i and an asset j, both the protocol fee and the asset fee come into play, since the outbound amount ΔQ_i in response to quantity ΔR_i in asset i swapped in is assessed the protocol fee f_Q before it is submitted as the inbound quantity ΔQ_j (which returns ΔR_j minus the protocol fee amount, as described above). The slippage in this case may be shown to be:

$$SL(i, j, f_Q, f_R) := \frac{\pi_j^i(f_Q, f_R)}{p_j^i} - 1 = \left(\frac{1}{(1 - f_R)(1 - f_Q)} - 1\right) + \frac{1}{1 - f_R}\left(\frac{1}{1 - f_Q} + \frac{Q_i}{Q_j}\right)\frac{\Delta R_i}{R_i}.$$

As expected, then, any amount of protocol and/or asset fees is deleterious to efficiency defined by slippage, and a trade-off thus exists between providing greater efficiency from smaller fees on the one hand, and greater impermanent loss mitigation to a liquidity provider from higher fees on the other.

Changes to Omnipool Specification

Colin Grove, HydraDX

November 17, 2022

This document outlines changes to the Omnipool Specification made *after* the Block Science report.

The changes outlined here were required due to issues noticed during the external audit of the Omnipool implementation. One issue was brought to our attention by the auditors; the other issue was noticed first by the HydraDX team.

The HydraDX team takes full responsibility for inclusion of the specifications *with* these issues in the Block Science report. We are very grateful to Block Science for their dedicated and thorough work, and we still believe the Block Science report has enormous value as an explanation and assessment of the Automated Market Maker (AMM) at the core of the Omnipool. We hope that this document can provide clarity and transparency regarding the relationship between the specification in the Block Science report, and the Omnipool's final implementation.

Block Science has *not* reviewed the specification changes described in this addendum, and the HydraDX team takes full responsbility for them.

Scope of changes

The changes involve two mechanisms that exist on top of the core Omnipool AMM: The TVL Cap, and the LRNA Imbalance parameter.

The core AMM (the path-independent Constant Function Market Maker) is unchanged. LRNA Imbalance calculations do occur during AMM transactions, but the only affect the LRNA Imbalance has on the core AMM is to determine whether the protocol fee is burned or sent to the HDX subpool. Likewise, the TVL Cap prevents the addition of liquidity beyond a certain TVL. Apart from causing add liquidity transactions that would violate the cap to fail, it does not affect the core AMM calculations.

Summary of reasons for changes

TVL Cap

During the audit of the Omnipool code, auditors noted an attack vector due to the TVL Cap mechanism as previously specified and implemented. HydraDX and auditors identified a way to simplify the TVL Cap mechanism so as to eliminate the attack vector, while retaining the desired TVL Cap functionality.

LRNA Imbalance

During the audit of the Omnipool code, an improper assumption was identified in the mathematical derivation of the specification of the LRNA imbalance. The assumption was simple to correct, but resulted in a slightly different mathematical specification. The purpose of the LRNA Imbalance did not change, and the implementation of the corrected specification was required to achieve this purpose.

Details

TVL Cap change

Problem with old specification In the old specification, the TVL of each asset, denominated in USD, was tracked as T_i. The variable T_i for asset i would only be updated upon the addition or removal of liquidity in asset i. When T_i was updated, the spot price of USD denominated in LRNA would be obtained from the Omnipool (which has a "preferred" stablecoin for this purpose) and this price would be used to calculate the TVL of asset i.

The spot price of any asset in any CFMM is easy to manipulate temporarily, and care must always be taken with mechanisms built on a spot price. The problem here arose because if an attacker manipulated the spot price of USD, and arbitragers moved the price back in line with external markets, any T_i calculated during the attack *would still be wrong* after the arbitrage trades restored correct spot prices. This led to a situation in which an attacker could cause the TVL cap to inappropriately block the addition of liquidity to the Omnipool.

Description of fix The fix is to simply calculate TVL direction from the total amount of LRNA in the pool, and the price of USD in LRNA, any time the TVL cap might be applied. With this fix, an arbitrager's trade which realigns the spot price of USD in the pool to external markets will also realign the price used in the calculation of TVL. An attacker may still move the spot price (indeed, all trades move spot prices) but they will only be able to potentially disable the addition of liquidity for the duration of the price divergence, and they will have to pay arbitragers every block to maintain the price divergence, making the attack prohibitively expensive.

LRNA Imbalance

Notation We will use Q_i to denote the quantity of LRNA in the asset i subpool, and R_i to denote the quantity of asset i itself. We will use L for the LRNA imbalance, and whenever a variable X has superscript $+$, it denotes the variable "after the transaction" being discussed. I.e., $X^+ = X + \Delta X$. Note that this also defines a sign convention for ΔX.

Problem with old specification The goal of tracking what we call the "LRNA Imbalance" is to track how much LRNA has been sold back into the Omnipool, and to burn LRNA fees to compensate for the depreciation of LRNA *due to* LRNA being sold back to the Omnipool.

The spot price of token i denominated in LRNA is $\frac{Q_i}{R_i}$.

The old specification was derived from the requirement that, during a LRNA swap (i.e. LRNA being sold back to the pool),

$$\frac{Q_i}{R_i} = \frac{Q_i^+ + L^+}{R_i^+}$$

We can think the right hand side as the *target price* that will be restored once LRNA is burned in the quantity L^+. It's important to know for the evaluation of this formula that $L < 0$. Thus L basically offsets the extra LRNA coming into the pool.

In the above equation, we set the target price (RHS) equal to the price before the transaction (LHS). However this is only the correct operation *if* $L = 0$ *before the transaction*, and additionally, if we expect all the extra LRNA to affect *only the subpool of asset* i.

The fix The proper setup is

$$\frac{Q_i + L\frac{Q_i}{Q}}{R_i} = \frac{Q_i^+ + L^+\frac{Q_i^+}{Q^+}}{R_i^+}$$

This setup fixes both issues: - The left hand side now reflects the *target price* before the transaction, even if $L < 0$ - The equation for target price now allocates the LRNA to be burned evenly across assets in the Omnipool, by weighting the affect of L on Q_i by the asset's weight $W_i = \frac{Q_i}{Q}$

Indeed, we can see that using the new target prices for assets j and i, we recover the *existing* spot price:

$$\frac{Q_i + L\frac{Q_i}{Q}}{R_i} \frac{R_j}{Q_j + L\frac{Q_j}{Q}} = \frac{Q_i}{Q_j}\frac{R_j}{R_i}\frac{1 + \frac{L}{Q}}{1 + \frac{L}{Q}} = \frac{Q_i}{Q_j}\frac{R_j}{R_i}$$

This demonstrates that the *new* target price formula is consistent with spot prices for non-LRNA assets.