Why We Can't Swap on Deposit or Withdraw

When depositing (or investing) in DeFindex, a user receives dfTokens, which represent their share of the DeFindex portfolio. These tokens are minted based on the amount of assets deposited and the current price per share (denoted as p_{ps}). Later, the user can burn these tokens to withdraw their assets.

The challenge arises when calculating this price per share.

Example Scenario

Consider a DeFindex with a single strategy: 100% allocation to XLM on Xycloans, while the DeFindex receives deposits in USDC. The price per share p_{ps} could be calculated as the amount of USDC one receives after withdrawing from Xycloans and swapping the XLM to USDC, divided by the total supply of dfTokens:

$$
p_{ps}(m) = \frac{p_{XLM}(m) \cdot M_{XLM}}{T_{dfTokens}}
$$

Where:

- $p_{XLM}(m)$ is the price of XLM in terms of USDC after liquidating m XLM.
- M_{XLM} is the total amount of XLM held by the DeFindex.
- $T_{dfTokens}$ is the total supply of dfTokens.

The problem is that the price of XLM $p_{XLM}(m)$ will depend on the amount of XLM we need to withdraw. This price can be manipulated by a large swap. For instance, someone could swap a large amount of USDC for XLM, artificially inflating the price of XLM. As a result, the price per share would increase, allowing the user to receive more USDC when burning their dfTokens.

Fixed Price Per Share Approach

PROF

Given the manipulation risk with a variable price per share, let's consider using a fixed price per share.

Assume p_0 is the nominal (initial or fixed) price of XLM in USDC. The amount of USDC received by a user who swaps m_{XLM} XLM for USDC will be:

$$
m_{USDC} = p_{XLM}(m_{XLM}) \cdot m_{XLM}
$$

The price per share would then be:

$$
p_{ps} = \frac{p_0 \cdot M_{XLM}}{T_{dfTokens}}
$$

After burning $m_{dfTokens}$ dfTokens, the user should receive:

$$
m_{USDC} = p_{ps} \cdot m_{dfTokens} = p_0 \cdot M_{XLM} \cdot \frac{m_{dfTokens}}{T_{dfTokens}}
$$

Where:

- m_{XLM} is the amount of XLM the DeFindex needs to liquidate to pay the user.
- $m_{dfTokens}$ is the amount of dfTokens the user is burning to withdraw their share.

The amount of XLM to be liquidated, m_{XLM}, is given by:

$$
m_{XLM} = M_{XLM} \cdot \frac{m_{dfTokens}}{T_{dfTokens}}
$$

The USDC received after the swap $m_{USDCout}$ would then be:

$$
m_{USDCout} = p_{XLM}(m_{XLM}) \cdot m_{XLM} = p_{XLM}(m_{XLM}) \cdot M_{XLM} \cdot \frac{m_{dfTokens}}{T_{dfTokens}}
$$

Since p_0 is the nominal price and $p_{XLM}(m_{XLM})$ is the actual price after liquidation:

$$
p_{XLM}(m_{XLM}) < p_0 \quad \forall m_{XLM} > 0
$$

Thus, we have:

$$
p_{XLM}(m_{XLM}) \cdot M_{XLM} \cdot \frac{m_{dfTokens}}{T_{dfTokens}} < p_0 \cdot M_{XLM} \cdot \frac{m_{dfTokens}}{T_{dfTokens}}
$$

PROF

In summary:

$$
m_{USDCout} < m_{USDC}
$$

This inequality shows that the user would request more USDC than what they can actually receive after the swap. This discrepancy leads to a potential loss of funds for DeFindex, highlighting why we can't rely on swapping assets during the deposit process.

We can argue that when using any fixed price per share, the amount of USDC received after a swap, denoted as $m_{USDCout}$, will differ from the expected amount m_{USDC}. This discrepancy introduces a vulnerability, making the protocol susceptible to manipulation.

Calculating Price Per Share (PPS):

The PPS is a crucial metric for ensuring users receive the correct value for their dfTokens. It is calculated as follows:

$$
\text{PPS} = \frac{\text{Total Assets}}{\text{Total Supply of dfTokens}}
$$

Where:

- **Total Assets**: The sum of the value of assets managed by all adapters plus any idle assets held directly by the DeFindex contract.
- **Total Supply of dfTokens**: The total number of dfTokens issued to users.

To illustrate, consider the following scenario:

- DeFindex has three adapters managing different investments:
  - Adapter A manages $50,000 in a liquidity pool.
  - Adapter B manages $30,000 in a lending pool.
  - Adapter C manages $20,000 in a staking protocol.
- The DeFindex contract holds an additional $10,000 in idle assets.

The Total Assets would be:

$$
50,000 + 30,000 + 20,000 + 10,000 = 110,000 \text{ USDC}
$$

If the Total Supply of dfTokens is 100,000, the PPS would be:

$$
\text{PPS} = \frac{110,000 \text{ USDC}}{100,000 \text{ dfTokens}} = 1.1 \text{ USDC per dfToken}
$$

This calculation ensures users can accurately determine the value of their holdings in DeFindex, promoting transparency and trust.