Construction Safety for “OutForIn” Functions

Created by	
Created at	@January 15, 2022 12:43 AM
Last updated by	
Last updated at	@January 31, 2022 10:31 PM

The task here is to prove that the checks at the end of the functions `hTokenOutForUnderlyingIn` and `underlyingOutForHTokenIn` are in fact superfluous. Specifically, these checks are:

```
if (hTokenReserves < newHTokenReserves) {
  revert YieldSpace__LossyPrecisionUnderflow(hTokenReserves, newHTokenReserves);
}
```

And:

```
if (normalizedUnderlyingReserves < newNormalizedUnderlyingReserves) {
revert YieldSpace__LossyPrecisionUnderflow(normalizedUnderlyingReserves, newNormalizedUnderlyingReserves);
}
```

For ease of calculation, we will use the following variables:

- $x_s =$ `normalizedUnderlyingReserves`
- $\Delta_x =$ `underlyingIn`
- $y_s =$ `hTokenReserves`
- $\Delta_y =$ `hTokenIn`
- $a = 1 - gt$, using the appropriate value of g (`G1` for `hTokenOUtForUnderlyingIn`, `G2` for `underlyingOutForHTokenIn`)

Note that the negation of Δ_x and Δ_y will represent `underlyingOut` and `hTokenOut`, respectively, for the appropriate functions.

The function `hTokenOutForUnderlyingIn` calculates the amount of hToken a user would receive for a given amount of underlying. This is represented mathematically as :

$$y_s - \Delta_y = (x_s^a + y_s^a - (x_s + \Delta_x)^a)^{1/a}$$

The value $y_s - \Delta_y$ is equal to the reserves of HToken after the trade. Assume by way of contradiction that `hTokenReserves < newHTokenReserves`. Then we would have:

$$y_s < y_s - \Delta_y \Rightarrow \Delta_y < 0 \Rightarrow -\Delta_y > 0$$

This would represent the amount of hTokenOut being positive, or in other words the contract receives underlyingIn from the user, while also adding more hToken to its reserves. This violates the YieldSpace equation, and so therefore $y_s \geq y_s - \Delta_y$ is always true, meaning that the condition `hTokenReserves < newHTokenReserves` is never satisfied.

A similar argument holds for `underlyingOutForHTokenIn` by switching the roles of Underlying and HToken.