Liquidation Calculations for Vaults

Kree, Isomorph loans

October 2022

1 Introduction

For liquidation we are initially looking to solve the following equation to find δx a small change in the collateral quantity which once sold would bring the users loan back up to the minimum margin ratio required.

$$P(x - \delta x) = (y - \delta y)L_m \tag{1}$$

We can rearrange this equation to make δx the subject and use the relationship that $\delta y = FP\delta x$ where:
$P :=$ price of collateral asset in USD
$x :=$ total user collateral amount
$y :=$ total user loan in USD
$L_m :=$ liquidation margin ratio
$F :=$ percentage of a loan that is returned to the system on liquidation.

All these quantities are known and so we can substitute and rearrange to find δx.

$$P(x - \delta x) = (y - FP\delta x)L_m \tag{2}$$

$$\frac{P(x - \delta x)}{L_m} = y - FP\delta x \tag{3}$$

$$\frac{Px}{L_m} - y = \frac{P\delta x}{L_m} - FP\delta x \tag{4}$$

$$\frac{Px}{L_m} - y = P\delta x(\frac{1}{L_m} - F) \tag{5}$$

$$\frac{\frac{Px}{L_m} - y}{P(\frac{1}{L_m} - F)} = \delta x \tag{6}$$

Equation 6 gives us the basic method of calculating how much of a given loan should be liquidated. However as we are only using integers in Solidity we then optimise the equation further to make it more suitable for this environment.

$F \in [0, 1]$ so cannot be directly used. We elect to expand the denominator bracket to solve this.

$$\frac{\frac{Px}{L_m} - y}{\frac{P}{L_m} - FP} = \delta x \tag{7}$$

While we can use negative numbers natively in Solidity doing so would reduce the range of numbers we could support which could lead to bugs or reverts. To solve this we multiple the entire left-hand side by -1.

$$\frac{y - \frac{Px}{L_m}}{FP - \frac{P}{L_m}} = \delta x \tag{8}$$

To make mistakes less likely we remove division from the equations by multiplying the LHS by L_m.

$$\frac{yL_m - Px}{FPL_m - P} = \delta x \tag{9}$$

We must be mindful that in Solidity if subtraction on unsigned integers would yield a negative number the code will instead revert. On the numerator this is not a problem as if $yL_m < Px$ then this suggests that liquidation would not be applicable as the user's collateral fully covers their current loan. So by checking this condition in viewLiquidatableAmount prior to subtraction we deduce if liquidation should be possible for the loan.
For the denominator the situation is less clear, so long as $FL_m \geq 1$ the equation will not revert due to a subtraction overflow. However as $F = 0.95$ in the current implementation this means $L_m \geq 1.053$. While this is a suitable bound it is something future developers/the DAO must be aware of when setting new collateral parameters.

Equation 9 now outlines the method we use in the function viewLiquidableAmount to calculate the quantity of collateral that should be sold in order to return a user's loan to a safe margin level. However several values used by the equation are decimals stored as integers by Solidity by multiplying by 10^{18}, the effected values are F, P and L_m. To adjust for this we use the constant LOAN_SCALE (LS for short) to adjust these values to their true values. Both numerator terms have been scaled up by 10^{18} so this factor can be removed by leaving a factor of 10^{18} in the denominator. The denominator's term FPL_m requires reduction by $(10^{18})^3$ while P requires reduction by 10^{18}. To adjust too early would lose accuracy so we reduce the first time by 10^{18} only and then shift P up by 10^{18} to bring them into the same scale. This means both terms now must be reduced by $(10^{18})^2$, which cancels to 10^{18} due to the reduction needed for the numerator. The final 10^{18} reduction is then done, while the equation could be written much cleaner it is left in this form to match the Solidity implementation.

$$\frac{(yL_m - Px)}{\frac{\frac{FPL_m}{LS} - PLS}{LS}} = \delta x \tag{10}$$