

# Blink Margin Engine

Proskurin Oleksandr

## Abstract

The paper describes a novel approach to calculating margin requirements for under-collateralized cross-chain leverage, considering DeFi-specific primitives such as liquidity pools, bridging, and concentrated liquidity. Key findings:

- 1) Blink Margin Engine is a centralized system designed to calculate portfolio risk using complex scenario analysis of the underlying asset prices.
- 2) The Engine is designed to provide portfolio risk analytics in a multi-chain environment.
- 3) Blink Margin Engine is designed to become a universal middleware and standard in DeFi margin calculation for trading operations.

## 1. Introduction

The recent massive explosion of DeFi activity and innovation led to decentralized lending becoming vital to an emerging financial system. Early DeFi's success has been driven by over-collateralized lending protocols such as Aave and Compound, which enable borrowers to deposit collateral in one token and withdraw part of its value in another. However, as the system becomes increasingly mature, on-chain market participants, especially asset managers, need access to undercollateralized loans, which provide more efficient capital allocation.

AAVE and Compound don't restrict operations with leverage users receive through their platforms. As a result, numerous restrictions need to be imposed on the platform. This leads to an inefficient relative size of a loan in relation to the value of the collateral - **overcollateralized loan** [6] [8]. Asset managers are willing to accept constraints on operations done with leverage (trading-specific only) in exchange for an undercollateralized, capital-efficient loan which gives the ability to provide a better return on invested capital. On the other hand, undercollateralized loans, if liquidated incorrectly, impose creditors on serious risks, which is why a sophisticated system (**Margin Engine**) is needed to define when a user's loan should be liquidated.

Secondly, the risks of several positions in a trader's portfolio may offset each other. For example, a portfolio with an ETH/USDC liquidity pool position hedged by short perpetual in ETH/USDT has a much lower risk than an unhedged LP

position in BTC/USDT or ETH/USDT. Margin calculations which account for total portfolio P&L (**Cross Margin**) instead of segregated risk calculation for each position (**Isolated Margin**) improve capital and leverage efficiency for professional asset managers as maintenance margin requirements in Cross Margin is much lower.

In this paper, we propose a unified framework for calculating cross-chain portfolio margin for under-collateralized loans taking into account De-Fi specifics:

- 1) Asset classes such as Liquidity Pools, Concentrated Liquidity LPs, and Liquid Staking Derivatives (LSD).
- 2) Automated Market Makers (AMM) and Price Oracles.
- 3) Gas costs and risks associated with Bridging.

## 2. Standard Portfolio Analysis of Risk (SPAN) and Standard Initial Margin Model (SIMM)

The problem of multi-asset portfolio risk calculation is a well-researched and known issue solved in the traditional finance space. SPAN was developed in 1988 by Chicago Mercantile Exchange Inc. (CME) to access risk on an overall portfolio basis effectively [3]. It calculates the potential loss in positions and sets this value as the initial margin payable by the firm holding the portfolio. As a result, SPAN provides for offsets between correlated positions and enhances margining efficiency. SPAN model has proven to be an efficient and robust way to calculate portfolio margin for multi-asset derivatives portfolios. It became the official Performance Bond mechanism of 54 exchanges and clearing organizations worldwide, making it the global standard for portfolio margining.

SPAN model relies on scenario analysis and asset price simulations, taking into account their correlation structure. Given that, positions P&Ls are calculated and aggregated on a portfolio level. Finally, SPAN groups financial instruments with the same underlying for analysis and estimates the portfolio's Value at Risk (VaR) metric, which serves as input for final margin calculations.

At the same time, the SIMM model developed by International Swaps and Derivatives Association (ISDA) is a common methodology for calculating the initial margin for uncleared OTC derivatives. It also relies on asset price simulations to

calculate potential portfolio risks and imposes several haircuts on the market values of riskier securities. [4]

In Blink, we use SPAN/SIMM scenario analysis of the underlying and haircuts approach as a starting point for Margin Engine design. As Blink Margin Engine relies upon heavy computations and scenario analysis, it is impossible to put these calculations on a blockchain, which is why it is a centralized system. In the meantime, it is still designed to be used as a standard in the DeFi space for decision-making regarding portfolio risk and maintenance margin calculation.

### 3. Architecture

Blink Margin Engine analyses the portfolio of whitelisted (supported) assets, tokens, and DEXs in real-time. The methodology is designed to be configured for any universe/Liquidation Routing/Asset Price Information. However, the paper will use a whitelisted universe from Blink Protocol. The list of supported assets, pools, and DEXs for Blink Protocol can be found in Appendix A.

The system is designed to give its user an answer to the following questions:

- 1) What is the current portfolio Net Asset Value (NAV) relative to a borrowed asset?
- 2) What is the stress-tested NAV? What is risk-factor and maintenance margin value right now?
- 3) What is the market impact/cost of liquidating token X for token Y on any supported DEX considering market impact?
- 4) What is the gas/bridging cost of liquidating a portfolio using defined Liquidation Routing?
- 5) What is the total liquidity pool position value relative to a borrowed asset considering fees to be claimed?
- 6) What actions and trades need to be completed to liquidate a portfolio considering the Liquidation Routing scheme if a portfolio should be liquidated right now?

#### 3.1. Reference Price and Liquidation Routing

The fundamental purpose of any margin engine is to calculate the initial and maintenance margin at any given time. If the trader's portfolio can not satisfy maintenance margin requirements - they get a margin call, and the portfolio is liquidated. As the liquidation process is quite specific in the DeFi space, especially in a multi-chain environment, market impact, gas, and bridging costs should also be included in margin calculations to detect liquidation levels.

First, we need to define **Reference Price** - the price of an asset X relative to the **Leverage Asset**. Blink Margin Engine

uses ChainLink Oracles for that. For example, if a trader took USDC leverage on Ethereum to open a leveraged position in ETH, then ETH/USDC (Ethereum) Chainlink Oracle is used to determine Reference Price at any given time. Secondly, we introduce **Liquidation Routing** - the weight of each DEX in the liquidation process if a margin call happens. Taking the example above, in Blink Protocol, for an open ETH position, if USDC is used as leverage, 70% of ETH will be converted back into USDC on Uniswap, and 30% will be swapped using 1INCH. We must swap each token from the user's portfolio during liquidation into a leverage asset to bring it back into the liquidity pool where it was taken\*. Reference Price and Liquidation Routing are used in **Mark Price** calculations.

#### 3.2. Mark Price for Position Value Calculations

Mark Price is used to get an estimate of currently open position values. As Mark Price estimates current position value considering unrealized profit and loss, it should consider current market liquidity for a specific token pair (asset/borrowed asset). That is why we introduce **Market Impact Factor**. Market Impact Factor estimates the market impact for any DEX from Liquidation Routing if a trader's position in token X is liquidated. That is why Market Impact Factor is a function of the:

- 1) DEX and current DEX liquidity.
- 2) Trader's position size in a token. Big, concentrated positions will have a bigger market impact.

At any given time  $t$  for a token  $i$ , Mark Price is

$$\text{Mark Price}_t^i = P_t^i \times [1 - \sum_{j=1}^N w_j \times f_t^j \times (1 + \text{MarketImpactPct})]$$

where

- $P_t^i$  - ChainLink Oracle price of a trading pair token/borrowed asset (Reference Price).
- $w_j$  - the percentage of DEX  $j$  in Liquidation Routing of a token  $i$  (for example, in Blink Protocol, for ETH, if the Borrowed asset is DAI, Uniswap weights 70%).
- $N$  - number of exchanges in Liquidation Routing for a token  $i$
- $f_t^j$  - Market Impact Factor for DEX  $j$  on token  $i$ .

To give even more margin of safety, Blink Margin Engine inflates Market Impact Factor calculations by  $\text{MarketImpactPct} = 15\%$  to consider unexpected market impact during liquidations.

As we have previously said,  $f_t^j = F(\tilde{E}, S, L_t)$  where  $\tilde{E}$  defines DEX and how execution is handled on a particular DEX (AMM, Order-Book, RFQ),  $S$  is the size of a position,  $L_t$  is liquidity on a given DEX at time  $t$ . Introducing the Market Impact Factor (MIF) allows us to estimate the cost of liquidation where Liquidation Routing may consist of both AMM and Order-Book-based exchanges.

\*The priority of liquidation is a position notion value (1st order of priority) and liquidity (2nd order or priority).

### 3.3. Estimating Market Impact Factor for Order-Book-based Exchanges

There are various well-known models [1], [5], [7] of quantifying market impact in order-book markets. However, none of them are used in Blink Margin Engine. The reason is that in market impact estimates, the Engine needs to answer, "By how much liquidating a large order size with a single market order will shift the price from the equilibrium?". This decision is mainly driven by the fact that the liquidation process on Blink Protocol is done through one order for each DEX in Liquidation Route. Once the liquidation process in the protocol becomes more advanced (splitting orders into smaller parts, VWAP, and other advanced execution algorithms), - market impact estimation will be changed accordingly.

Blink Margin Engine uses a "walking over" limit order book approach to estimate Market Impact Factor. We use 5 levels of the limit order book to calculate the average execution price:

$$\text{Average Execution Price}_t = \frac{\sum_{i=1}^5 (P_i \times Q_i)}{\sum_{i=1}^5 Q_i}$$

In this formula,  $P_i$  represents the price at which the market order was executed at level  $i$  of the limit order book, and  $Q_i$  represents the number of shares executed at that level. The summation is taken over all  $n$  levels of the limit order book that were walked over by the market order. The numerator calculates the total value of shares executed at each price level, and the denominator calculates the total quantity of shares executed. The resulting value gives the average price per share at which the market order was executed.

$Q_i$  depends on liquidity at a particular level and liquidity on previous levels of the limit order book. For example, suppose we have the following limit order book for ETH/DAI pair shown in Table 1.

Buy orders	Sell orders
100 at 1000.00	200 at 1005
50 at 995	150 at 1010
75 at 990	100 at 1020
25 at 985	50 at 1030
50 at 980	25 at 1035

Now let's consider a market order to sell 150 ETH. The market order "walks over" the limit order book, executing at each level until filled. The first 100 ETH of the market order will fill the first level of the buy side of the order book for 1000.00, which exhausts this level. The remaining 100 ETH of the market order will then fill the next level of the buy side of the order book for 995, which exhausts this level of the order book. The market order is now filled, having sold 150 ETH at an average executing price of:

$$\frac{(100 \times 1000) + (50 \times 995)}{150} = 998.3$$

So the average executing price of the market order that walked over levels of the limit order book was 998.3.

Finally, the Market Impact Factor is the absolute percent difference between Average Execution Price and Mid-Price,  $V_t = (bid_t + ask_t)/2$ :

$$f_t = \left| \frac{V_t}{\text{Average Execution Price}_t} - 1 \right|$$

In our example, the Market Impact Factor equals  $\left| \frac{(1000+1005)/2}{998.3} - 1 \right| = 0.0042$

### 3.4. Estimating Market Impact Factor for Uniswap V3

Instead of using analytical formulas to estimate the market impact of executing a big order through DEX, Blink Margin Engine applies "smart contract polling" approach. The idea is to call DEX-specific smart contracts to see the expected price impact. In Uniswap V3, the Engine uses Uniswap V3 SDK, and the *UniswapV2Router02* contract deployed at `0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D` to estimate the price impact of a specific trade in a given pool. [2]

### 3.5. Estimating Market Impact Factor for Curve

For Curve, the Engine again utilizes Curve Finance SDK. The SDK provides for order router function call (*curve.router.ExchangeExpected*) to provide the expected number of tokens considering a trading pair and quantity.[2]

### 3.6. Estimating Market Impact Factor for DEX Aggregators(IINCH)

Aggregators are the DeFi equivalent of smart order routing systems. If an asset trades in multiple pools or DEXs, aggregators will buy the token across all pools to minimize the price impact on each. Instead of spreading the trade over time in a single market, this order executes at once, spread over many possible markets. [9] IINCH SDK provides a user with the *getQuote* function, which calculates the estimated price impact of a trade based on the current state of the IINCH DEX aggregator.

### 3.7. Mark Price Calculation Example

Imagine a trader borrowed 2000 USDT to buy 1 ETH. At the time  $t_1$ , the Reference Price of ETH/USDT is 1000 USDT. Using the Liquidation Routing from Blink Protocol, we can see that if liquidated:

- 70% of ETH will be liquidated through Uniswap.
- 30% of ETH will be liquidated through IINCH.

Using the algorithms described in 3.4 and 3.6, we may define the market impact factor for liquidating:

- 0.7 ETH on Uniswap is 0.03%.
- 3.7 ETH on IINCH is 0.01%.

Which results in a Mark Price of ETH relative to USDT:

$$1000 * (1 - 0.7 * 0.0003 * 1.15 - 0.3 * 0.0001 * 1.15) = 999.724$$

## 4. Portfolio Value Calculations

The first step in calculating a portfolio maintenance margin and a liquidation risk factor is to estimate each portfolio position's value separately.

#### 4.1. Token Price Calculations

For whitelisted tokens, we use Mark Price to estimate the price of a token relative to a borrowed asset. However, several tokens have a **valuation haircut**:

- stETH, wstETH: -5%
- CRV, cvxCRV: -15%

For example, if the Mark Price of CRV/USDT is 1.088 and a user has 100 CRV in their portfolio in Blink Finance, the value of the position will be:

$$1.088 * 100 * (1 - 0.15) = 92.48$$

#### 4.2. Liquidity Pools Calculations

Liquidity pools consist of staked assets and fees (rewards) to be claimed. If a user claims fees - their price is calculated by the Token Price Calculations method only if a fee token is whitelisted inside Blink Margin Engine. Non-whitelisted tokens are not included in portfolio value calculations. Staked assets are calculated using the Token Price Calculations methodology.

Unclaimed rewards (if rewards tokens are whitelisted by Blink) are calculated using the Token Price Calculations algorithm with a **haircut of 50%**.

For example, if a user has Uniswap V3 position with **5 ETH and 10 000 USDT** and **0.7 ETH and 100 USDT in unclaimed rewards** to be claimed with the borrowed asset is USDT, and ETH/USDT Mark Price is 1000 then its position value is:

$$(5 * 1000 + 10000) + (0.7 * 1000 + 100) * 0.5 = 15400$$

#### 4.3. Lending Protocols Calculations

For positions where a trader deposits collateral into a lending protocol (AAVE, Compound), a position value is calculated using Token Price Calculations by mapping the number of protocol tokens (aWETH, aUSDT, cUSDC, etc.) to collateral value.

For borrowed assets, the position value is calculated as follows:  
 $\text{Position Value}_t = \text{Collateral Deposited Value}_t$

$$- \text{Borrowed Asset Value}_t - \text{Accrued Interest}_t$$

For example, if a trader used Blink Protocol USDT leverage to open a short position in 1 ETH through AAVE, the algorithm is following:

- 1) Use borrowed from Blink 1600 USDT as collateral to borrow 1 ETH from AAVE. (1 ETH = 1000 USDT).
- 2) Sell immediately 1 ETH for 1000 USDT.

At time  $t_0$  the value of the portfolio is:  $1600 + 1000 - 1 * 1000$ .  
If the price of ETH at  $t_1$  becomes 800 the value of the portfolio is:

$$1600 + 1000 - 1 * 800 = 1800$$

On the other hand, an increased ETH price reduces a portfolio value, moving a trader closer to the liquidation point.

#### 4.4. Aggregating Portfolio by Assets

As previously discussed, position value significantly influences the Market Impact Factor in Mark Price calculations, so we must aggregate portfolio positions by tokens.

So, the position value in a token is the sum of all tokens on all chains, tokens in liquidity pools, and fees to be claimed in this token. For example, a trader has a portfolio built by borrowing USDC:

- 10 wETH, 5 wBTC on Ethereum.
- 7 wETH, 10 000 USDT on BSC.
- Uniswap V3 position (10 wETH x 6000 DAI) + fees (3 wETH, 200 DAI) on Ethereum.
- Curve LP position in (5000 USDT/5000 USDC) and fees to be claimed (100 CRV, 200 USDT, 5000 USDC).

As a result of aggregation, we have:

- wETH:  $10+7+10+3 = 30$  wETH.
- wBTC: 5 wBTC.
- USDT:  $10000 + 5000 + 200 = 15200$  USDT.
- DAI:  $6000 + 200 = 6200$  DAI.
- USDC:  $5000 + 5000 = 10000$  USDC.
- CRV: 100 CRV.

The aggregated token values are now used as inputs to calculate the Market Impact Factors. As we can see, after aggregation, we have a relatively big position in wETH, which is why the Market Impact Factor will influence the Mark Price heavily due to its significant value.

For borrowed asset in a portfolio (USDC in the example above), MIF equals 0, as no swap activities are needed during liquidation. Only a return to LP is required.

### 5. Adjusting Portfolio Value for Gas/Bridging Fees

This part of portfolio risk factor calculation applies only if the protocol/leverage entity intends to liquidate bad debt positions. In Blink Protocol, for example, the liquidation is done through Liquidation Routing without relying on external liquidators. The adjustment is needed as gas and bridging costs may significantly impact the overall liquidation cost. There are 4 types of actions that incur gas costs:

- Asset swap (when we need to swap assets back to borrowed asset).
- Extract liquidity from a pool/close debt on a lending protocol.
- Bridging.
- Return funds to LP.

For tokens bridging fees arise if a token is not on the chain defined by the Liquidation Routing. For pools, gas fees for extracting liquidity/claiming fees are added.

From the example above:

- Bridge+gas fees: 7wETH, 10 000 USDT from BSC to Ethereum.
- Gas fees: extracting liquidity from Uniswap V3, Curve.
- Gas fees: swapping 30 wETH, 5wBTC, 15200 USDT, 6200 DAI, 100 CRV.
- Gas fees: return USDC to Liquidity Pool.

Finally, the process of Portfolio Value Calculation can be described as follows:

- 1) Aggregate portfolio positions by tokens.
- 2) Define the Market Impact Factor as a function of position value for each token in a portfolio.
- 3) Get tokens Mark Prices.
- 4) Get the sum of token prices/liquidity pools/fees prices using Mark Prices and apply haircuts where needed.
- 5) Get the sum of assets and adjust it for gas/bridging fees.

## 6. Portfolio Stress-Test Scenario Analysis

Before calculating the **Risk Factor** used to determine if a user should be liquidated, we need to calculate **Stress Tested Portfolio Value**. Lets use the example from above to show how calculations work in this case. A user has a portfolio with borrowed asset USDC:

- 10 wETH, 5 wBTC on Ethereum.
- 7 wETH, 10 000 USDT on BSC.
- Uniswap V3 position (6.22 ETH x 8887 DAI in the range [700, 1300]) + fees (3 ETH, 200 DAI) on Ethereum.
- Curve LP position in (5000 USDT/5000 USDC) and fees to be claimed (100 CRV, 200 USDT, 5000 USDC).

Mark Prices are:

- wETH/USDC: 1000
- wBTC/USDC: 10000
- DAI/USDC: 0.99
- USDT/USDC: 0.99
- CRV/USDC: 1.08

For all assets/LP positions held in the user's portfolio, we want to understand **What happens to a position if its Mark Price drops by X% relative to the Leverage Asset?**.

### 6.1. Prepare Scenario Analysis Matrix

We stress-test available assets based on either exchange rate risk or de-pegging (for stablecoins, ETH/stETH) risk. Based on a token risk and borrowed asset, we apply a different rate of change in the scenario matrix. **Scenario Matrix,  $\Omega$** , is the matrix where each element represents a % change drop of the price of asset  $i$  relative to borrowed asset  $j$ . Scenario Matrix is a configurable parameter of Blink Margin Engine; in our example, we will use the matrix from Blink Protocol (Appendix B).

### 6.2. Get stress-tested values for portfolio tokens

Based on borrowed assets and Mark Price for each token from the users portfolio, we get the token value using Mark Price adjusted for the value from the Scenario Matrix. The stress-tested portfolio tokens values in USDC are:

- 10 wETH [10 \* 1000 \* (1-0.3) = 7000], 5wBTC [5\*10000\* (1-0.3) = 35000] on Ethereum.
- 7 wETH [7 \* 1000 \* (1-0.3) = 4900], 10 000 USDT [10000 \* 0.99 \* (1-0.1) = 8910] on BSC.

### 6.3. Get stress-tested values for LP positions

For LP positions, we need to consider not only delta risk (change of an asset price) but also gamma risk (Impermanent Loss). Taking the Impermanent Loss into consideration is extremely important, considering the non-linear nature of Liquidity Pools, especially those exposed to concentrated liquidity (Uniswap V3). When stress-testing, LP positions the algorithm in the following:

- 1) Calculate the updated token price ratio considering the updated exchange rate relative to a borrowed asset.
- 2) Get updated LP position quantities under the new token X/token Y price ratio.
- 3) Get updated price of LP position denominated in a borrowed asset.

#### 6.3.1. Estimating position quantities for Uniswap V2

Using the constant product formula  $x * y = k$  where  $x$  is the number of tokens X in the pool and  $y$  is the number of token Y, we can get the estimate of expected token quantities inside of LP as a function of  $k$  and the exchange rate  $p_t$  between  $x$  and  $y$ :

$$x_t = \sqrt{k/p_t}$$

$$y_t = \sqrt{k * p_t}$$

Secondly, for each LP position, we keep the percentage stake,  $s_t$ , in a pool for a given position reflected in "liquidity tokens" provided to a user. If others subsequently add/withdraw coins, new liquidity tokens are minted/burned, so everyone's relative percentage share of the liquidity pool remains the same. [10].

Finally, a stress-tested number of tokens for a user  $(x^*, y^*)$  is:

$$x^* = s_t * \sqrt{k/p^*}$$

$$y^* = s_t * \sqrt{k * p^*}$$

Where  $p^*$  is the price ratio of stress-tested token X to token Y prices:

$$p^* = \frac{\Omega(\text{borrowed asset, token X})}{\Omega(\text{borrowed asset, token Y})}$$

#### 6.3.2. Estimating values for Uniswap V3

For concentrated liquidity, price ranges may significantly impact updated token quantities under a stress-test scenario. As a result, Uniswap V3 LP positions are exposed to a bigger gamma risk than V2. We use Uniswap V3 whitepaper to estimate the updated values of LP positions on a price change. [11].

Consider,  $x_0, y_0$  are the initial token amounts deposited into LP in a price range  $[u, l]$  where the price ratio is  $p_0 = \frac{x_0}{y_0}$ . First, we need to calculate liquidity values for each of the tokens. According to Uniswap V3 whitepaper, we need to set token decimals constants  $(d_x, d_y)$ , for example, in WETH/USDC pool token decimals equal 18 and 6, respectively. The liquidity formulas for X and Y tokens are:

$$L_x = \frac{x_0 \cdot 10^{d_x} \cdot \sqrt{p_0 \cdot 10^{d_y - d_x}} \cdot \sqrt{u}}{\sqrt{u} - \sqrt{p_0}}$$

$$L_y = \frac{y_0}{\sqrt{p_0} - \sqrt{l}} \cdot \frac{1}{\sqrt{10^{d_y - d_x}}} \cdot 10^{d_y}$$

When  $L_x, L_y$  are calculated, we get  $L_{min} = \min(L_x, L_y)$  the following formulas can be used to get the updated token amounts  $(x^*, y^*)$ :

$$x^* = \frac{L_{min}}{10^{d_x}} \cdot \frac{\sqrt{u} - \sqrt{p^*}}{\sqrt{u} \cdot \sqrt{p^*}} \cdot \frac{1}{\sqrt{10^{d_y - d_x}}}$$

$$y^* = \frac{L_{min} \cdot (\sqrt{p^*} - \sqrt{l})}{10^{d_y}} \cdot \sqrt{10^{d_y - d_x}}$$

Where  $p^*$  is the updated price ratio.

Let's consider the following example: WETH/USDC LP position, which consists of 2.4 WETH and 3981 USDC provided in a price range of [1442.36, 1889.42] if the pool price of WETH/USDC is 1651.35. What would be the token prices if the price increases to 1739.5?

$$L_x = \frac{2.4 * 10^{18} * \sqrt{1651.35 * 10^{6-18}} * \sqrt{1889.42}}{\sqrt{1889.42} - \sqrt{1651.35}} = 1.4976 * 10^{15}$$

$$L_y = \frac{3981}{\sqrt{1651.35} - \sqrt{1442.36}} * \frac{1}{\sqrt{10^{6-18}}} * 10^6 = 1.4975 * 10^{15}$$

$$L_{min} = 1.4975 * 10^{15}$$

Finally, new token values are:

$$x^* = \frac{1.4975 * 10^{15}}{10^{18}} * \frac{\sqrt{1889.42} - \sqrt{1739.5}}{\sqrt{1889.42} * \sqrt{1739.5}} * \frac{1}{\sqrt{10^{6-18}}} = 1.4539$$

$$y^* = \frac{1.4975 * 10^{15} * (\sqrt{1739.5} - \sqrt{1442.36})}{10^6} * \sqrt{10 * 6 - 18} = 5584$$

In our example, for a liquidity pool of 6.22 ETH and 8887 DAI, and an initial price of 1 ETH = 1000/0.99 DAI, we can find the updated quantities if the stress-tested price (from Scenario Matrix) of a pool is:

$$p^* = 1000 * 0.7 / (0.99 * 0.9) = 785.634$$

In this case, the stress-tested position values are 13.25 WETH and 2623 DAI, and the total-stress-tested value of the LP position (excluding fees) is  $13.25 * 700 + 2623 * 0.891 = 11612$  USDC.

#### 6.4. Get stress tested fees values

For token values, we use the same approach as we did with tokens + haircut adjustment.

- Uniswap V3 fees: 3 ETH, 200 DAI =  $3 * 700 * 0.5 + 200 * 0.891 * 0.5 = 1139$  USDC.
- Curve pool fees: 100 CRV, 200 USDT, 5000 USDC =  $100 * 0.756 * 0.5 + 200 * 0.891 * 0.5 + 5000 * 0.5 = 2626.8$  USDC.

#### 6.5. Find the total positions sum and adjust for gas/bridging fees

Once we get adjusted for stress test position values in borrowed asset, we find the total sum and adjust for the value for gas/bridging fees.

### 7. Risk Factor Calculation

Finally, we can calculate the **Risk Factor** used to determine if a traders position should be liquidated.

$$RiskFactor_t = \frac{StressPortfolioValue_t - LiquidationFee}{BorrowedAssetAmount}$$

$$\frac{- LiquidationPremium - Interest(t)}{BorrowedAssetAmount}$$

**Liquidation Premium** is an extra margin of safety Blink Protocol takes to liquidate a user for unpredicted events which may occur, while **Liquidation Fee** is a fee that is deposited to populate Blink Insurance Fund. In Blink Protocol, *LiquidationPremium* and *LiquidationFee* are set to 10%. *Interest(t)* refers to cumulative accrued interest for using leverage. In our example, if gas/bridging fees are 2000, the USDC stress-tested portfolio value is:

$$35000 + 8910 + 11612 + 9500 + 1139 + 2628 - 2000 = 66789$$

Suppose, *Interest(t)* is 100, *BorrowedAssetAmount* is 20000 USDC then:

$$RiskFactor_t = \frac{66789 * (1 - 0.1 - 0.1) - 100}{20000} = 2.6656 > 1$$

A user is liquidated if their Risk Factor **drops below 1**. As a result, the maintenance margin is the amount of margin such that the Risk Factor equals 1.

### 8. Conclusions

The paper describes the process of calculating maintenance margin requirements using scenario analysis, Reference Price, Liquidation Routing, Mark Price, and Market Impact Factor to manage risk in a cross-chain DeFi ecosystem. The article provides a detailed explanation of these concepts and metrics, including how to estimate the Market Impact Factor for different types of DEXs and perform scenario analysis of constant-product and concentrated liquidity AMMs.

Further research will include building a more sophisticated Liquidation Routing system, which should be more robust and decrease market impact and slippage.

### References

- [1] Robert Almgren and Neil Chriss. Optimal trading with stochastic liquidity and volatility. *Journal of risk*, 7(3):5–19, 2005.
- [2] Curve Finance. Curve-js api documentation, 2023.
- [3] CME Group. Span margin methodology, 2021.
- [4] ISDA. Standard initial margin model (simm), 2016.
- [5] Robert Kissel. The science of algorithmic trading and portfolio management. *The Journal of Trading*, 4(4):8–13, 2004.
- [6] Stani Kulechov and Aave Community. Aave: A decentralized, open source, and non-custodial liquidity protocol, 2020.
- [7] Albert S Kyle. Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society*, pages 1315–1335, 1985.
- [8] Robert Leshner and Compound Labs. Compound: The money market protocol, 2018.
- [9] Dan Robinson, Alex Evans, and Tarun Chitra. Understanding amm price impact, 2020.
- [10] Uniswap. Advanced topics in uniswap v2, 2021.
- [11] Uniswap Labs. Uniswap V3: Concentrated Liquidity. <https://uniswap.org/whitepaper-v3.pdf>, 2021.

## Appendix A

### Blink Protocol Whitelisted Assets/Pools/DEXs

Whitelisted tokens:

- wETH
- wstETH, stETH
- wBTC
- DAI
- USDT
- USDC
- FRAX
- CRV, cvxCRV
- FXS
- LIDO
- MATIC

Leverage tokens:

- wETH
- wstETH
- wBTC
- DAI
- USDC
- FRAX

Whitelisted DEXs(Pools):

- Uniswap V2/V3: DAI-USDC, WBTC-WETH, USDC-WETH, USDC-USDT.
- Curve/Convex Finance/Frax Convex: ETH-wstETH, DAI-USDC-USDT, ETH-USDT-wBTC, CRV-ETH, FRAX-USDC, DAI-FRAX-USDC-USDT, CRV-cvxCRV (Curve only).
- 1INCH (swaps): ETH, wstETH, WETH, DAI, USDC, USDT, WBTC, CRV, CVX, FXS, MATIC, LIDO.
- Pancake Swap: USDT-USDC
- AAVE: ETH(wETH), WBTC, USDT, USDC, DAI, wstETH.

## Appendix B

### Scenario Analysis Matrix

	wETH(ETH)	wstETH	wBTC	DAI	USDT	USDC	FRAX	CRV	FXS	LIDO	MATIC
wETH(ETH)	-	-10%	-10%	-30%	-30%	-30%	-30%	-30%	-30%	-30%	-30%
wstETH		-	-30%	-30%	-30%	-30%	-30%	-30%	-30%	-30%	-30%
wBTC			-	-30%	-30%	-30%	-30%	-30%	-30%	-30%	-30%
DAI				-	-10%	-10%	-10%	-30%	-30%	-30%	-30%
USDT					-	-10%	-10%	-30%	-30%	-30%	-30%
USDC					-10%	-	-10%	-30%	-30%	-30%	-30%