

Arkis Margin Engine

Proskurin Oleksandr, Serhii Tyshchenko

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) Arkis 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) Arkis 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** [3] [4]. 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 [1]. 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. [2]

In Arkis, we use SPAN/SIMM scenario analysis of the underlying and haircuts approach as a starting point for Margin Engine design. As Arkis 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

Arkis 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 Arkis Protocol. The list of supported assets, pools, and DEXs for Arkis 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**. Arkis 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 Arkis 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. Mark Price is a proxy to the fair value of an asset taking into account current market liquidity, execution constraints, and other factors. For example, one may adjust Reference Price for implied market impact if the trader's position may be liquidated in the future. In case of Arkis Protocol and Margin Engine, **Mark Price equals Reference Price**.

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.

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

$$1.088 * 100 * (1 - 0.15) = 108.8$$

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 Arkis Margin Engine. Non-whitelisted tokens are not included in portfolio value calculations. Staked assets are calculated using the Token Price Calculations methodology.

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:

$$(5 * 1000 + 10000) + (0.7 * 1000 + 100) = 15800$$

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

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 Arkis Protocol USDT leverage to open a short position in 1 ETH through AAVE, the algorithm is following:

- 1) Use borrowed from Arkis 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.

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 Arkis 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) Get tokens Mark Prices.
- 3) Get the sum of token prices/liquidity pools/fees prices using Mark Prices and apply haircuts where needed.
- 4) 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, Ω** , 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 Arkis Margin Engine; in our example, we will use the matrix from Arkis 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. [5].

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. [6].

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 = $3 * 700 + 200 * 0.891 = 2278.2$ USDC.
- Curve pool fees: 100 CRV, 200 USDT, 5000 USDC = $100 * 0.756 + 200 * 0.891 + 5000 = 5253.6$ 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 Arkis Protocol takes to liquidate a user for unpredicted events which may occur, while **Liquidation Fee** is a fee that is deposited to populate Arkis Insurance Fund. In Arkis 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. Initial Margin Calculation

The initial margin needed to open the position corresponds to a value under which portfolio Risk Factor > 1 . However, as the trader decides which tokens to use as collateral (several tokens can be used), the problem of initial margin calculation is a function of:

- 1) Leverage asset and leverage amount.
- 2) Input tokens and token amounts.
- 3) Scenario analysis matrix values.

For token-agnostic, the initial margin (IM) value in the borrowed token is:

$$IM = Borrowed\ Amount * (0.3 + 0.1 + 0.1)$$

where 0.3 corresponds to the minimum value of scenario stress test in Scenario Analysis Matrix, Ω , 0.1 and 0.1 correspond to *LiquidationFee* and *LiquidationPremium* values.

For example, if a trader aims to borrow 2 000 000 USDC from Arkis Protocol, the token-agnostic initial margin amount is

$$2\ 000\ 000 * 0.5 = 1\ 000\ 000\ USDC$$

However, if an asset manager aims to provide the collateral in stable coins (USDT, FRAX, or DAI), the stress-tested value is 10% instead of 30%, which leads to a lower amount of initial margin to be deposited. That is why Arkis Margin Engine provides the module to calculate the **Initial Collateral Portfolio**, which is the portfolio of assets used to open a leveraged position such that the initial margin requirements are satisfied (Risk Factor > 1).

8.1. Initial Collateral Portfolio Calculation

Suppose a trader has whitelisted assets that can be potentially used as collateral defined as $T_1 \dots T_N$. Additional to that, for each asset T_j a trader sets maximum and minimum dollar amount (d_j^{max}, d_j^{min}) such that:

$$d_j^{min} \leq d_j \leq d_j^{max}, j = 1 \dots N$$

Finally, we need to find a minimum value portfolio which satisfies the above constraints, with Risk Factor > 1 . The resulting solution will give us token amounts which sum up in a portfolio that satisfies constraints and yields the most efficient leverage allocation for a trader. Let's denote

- p_j as Mark Price of asset j relative to a borrowed asset.
- q_j is the quantity of token j .
- $RF(q_t, \dots, q_n | B)$ is a Risk Factor of portfolio consisting token of amounts q_t, \dots, q_n and borrowed amount B .
- $d_j = p_j * q_j$ is the dollar value of asset j in Initial Collateral Portfolio.

In this case, the optimization problem is:

$$\begin{aligned} & \underset{q_1, \dots, q_N}{\text{minimize}} && \sum_{j=1}^N p_j q_j \\ & \text{subject to} && d_j^{min} \leq p_j q_j \leq d_j^{max}, j = 1, \dots, N, \\ & && RF(q_1, \dots, q_N | B) > 1. \end{aligned}$$

One should take into account that RF is a function of stress-tested prices from Ω .

9. 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] CME Group. Span margin methodology, 2021.
- [2] ISDA. Standard initial margin model (simm), 2016.
- [3] Stani Kulechov and Aave Community. Aave: A decentralized, open source, and non-custodial liquidity protocol, 2020.
- [4] Robert Leshner and Compound Labs. Compound: The money market protocol, 2018.
- [5] Uniswap. Advanced topics in uniswap v2, 2021.
- [6] Uniswap Labs. Uniswap V3: Concentrated Liquidity. <https://uniswap.org/whitepaper-v3.pdf>, 2021.

Appendix A

Arkis 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%