

# Frankencoin\*

Luzius Meisser\*\*  
*Meisser Economics*

Basile Maire\*\*\*  
*Desma Eight, LLC*

January 11, 2022

---

\*Preliminary

\*\*[luzius.meisser@gmail.com](mailto:luzius.meisser@gmail.com)

\*\*\*[basile.maire@desma8.io](mailto:basile.maire@desma8.io).

# Frankencoin

## Abstract

blabla

*JEL Classification Codes: D40, G23*

# 1 Introduction

We start from the use-case of a Swiss Frank Lombard loan, collateralized with tokens such as Bitcoin, or tokenized shares. The borrower deposits collateral into the system and thereby mints a token, the “Frankencoin” ZCHF, that is pegged to the Swiss Frank. If the value of the borrower’s collateral falls below a certain threshold, the loan can be liquidated and the borrower incurs a haircut. If the value of the collateral falls below the loan value before the liquidation ends, the loss eats into capital. Implementing this system in a fully decentralized way on the Blockchain leads us to our main contribution.

First, we introduce an algorithmic stablecoin that allows for *different collateral types and liquidation rules*. This diversification of collateral and liquidation rules reduces the impact of non-systemic market events to the stablecoin peg. Second, we propose a specific mint plugin setup for which

- We use an *auction mechanism* that avoids using external Oracle price feeds
- To calibrate the system, we apply traditional risk management techniques that have not found their way into the world of stablecoins

The system entails a decentralized governance that governs risk parameters and can deny the community’s addition of new peg-methodologies if deemed unfit.

The Frankencoin can be minted by anyone using one of the available mint plugins. We define mint plugins as smart contracts that have the ability to mint Frankencoin in accordance with their rules. Mint plugins have to be approved by the governance mechanism and there might be different types of mint plugins for different types of collateral and different liquidation mechanisms. Anyone can propose to add a new mint plugin and if it is approved by the governance mechanism, it can be used according to its rules. These rules can include interest rates that are owed by the minters, liquidation mechanisms, collateral requirements, and required ZCHF to be held by “stakers” as a reserve per ZCHF token minted. While every mint plugin defines their required reserve capital, the reserves are shared among all mint plugins.

Stakers are participants that lock their Frankencoin into the system. Stakers earn interest rates paid for by borrowers and in turn risk to lose Frankencoin in case the liquidation process is not able to recover the loan amount with the available collateral. Staked Frankencoin are subject to a lockup period. This ensures that in case of a "bank run" on the system, stakers are the last who can liquidate their Frankencoin and will therefore also suffer from the greatest losses while the other participants are likely to be able to reclaim the full value.

This paper focuses on two aspects of Frankencoin. First, we discuss how the Frankencoin embeds with the wider economy. Many existing stablecoin systems fall short of this aspect and can arguably only be sustained with significant growth of the system or demand of their own token, see e.g., [?]. We propose a specific setup of mint plugins for which we use standard risk methodology techniques to quantify adverse events and calibrate capital reserves and fees. This is in stark opposition to existing stablecoins that set parameters such as the collateralization-ratio without risk methodological quantification methods. *TODO: reference??*

## 2 Economy

The immediate participants of Frankencoin consist of *minters*, *users*, and *stakers*.

*Minters* deposit collateral into a mint plugin and thereby enter a Frankencoin Lombard loan. Minters benefit from the system to the extent that it allows them to use liquid or illiquid assets that they want to hold long-term, to generate short-term liquidity. This can be motivated e.g., by tax considerations in some jurisdictions, or by justifications that hold for traditional Lombard loans. Similar to Lombard loans, the position is overcollateralized.

*Stakers* stake Frankencoin in a dedicated contract, thereby locking these funds up for a certain period of time. As a consequence, stakers are the last to be able to liquidate their holdings in case of a collapse of the value of Frankencoin. Stakers have to be compensated for their risk. This compensation is provided by the minters. Fees and interest rates paid by the minters are distributed to the stakers. By staking Frankencoin, stakers release governance tokens that give them voting rights.

*Users* hold and transfer Frankencoin as a means of payment or store of value. No fees are charged to the users, but they also are not provided with any financial gain from holding Frankencoin by the system. The system should be designed such that the tail risk of a complete default is negligible for the users as that risk is outsourced to the stakers.

This setup separates risk-takers (stakers) from the users. Governance token holders have "skin in the game" and are thus incentivized to maintain a healthy system. For instance, because staker capital is shared for all mint plugins, stakers have a vested interest to retain a healthy ecosystem of mint plugins.

Creditors of Frankencoin loans should be paid a risk-free rate corresponding to the Swiss Frank, plus a risk-premium. Otherwise, there is no rationale why stakers should sustainably lock their funds in Frankencoin (other than for non-pecuniary reasons). From the perspective of the minters, this means that costs to minters should be in line with the market capital costs for their loan.

We now address the conditions under which the *peg to the Swiss Frank* should hold. We approach the valuation of the Frankencoin from the perspective of a *perpetual bond*. A perpetual bond, or consol, is a bond with coupon payments but no redemption date, see, e.g., [?]. The staked Frankencoin is subject to default risk, because the system burns ZCHF when a minter's position is undercollateralized. We price this credit-risky perpetual along the lines of [?], by discounting the interest payments on a credit-risky term structure. Let's assume that interest payments happen at discrete time-steps  $0, \dots, \infty$  and we have corresponding risky rates of the term-structure so that the date-0 value of a promised Swiss Franc at time  $t$  of a credit-risky Franc promise is equal to  $\exp(-r_t t)$ . Let the constant coupon rate per Frankencoin be  $c$ . Now, the value of the perpetual can be written as

$$v(0) = \sum_{t=0}^{\infty} c e^{-r_t t} \tag{1}$$

$$= \sum_{t=0}^{\infty} c e^{-y t} \tag{2}$$

$$= \frac{c}{1 - e^{-y}}, \tag{3}$$

where the second line replaces the time-specific discount rates by a yield, and the last line is an application of geometric series. For the value to be at par,  $v(0) = 1$ , we have to choose the coupon rate accordingly:  $c = 1 - e^{-y}$ . Hence, if the interest earned from staking ZCHF are in line with discounting, the present value of one ZCHF is equal to one Swiss Franc.

The credit risky term-structure corresponds to the Swiss Franc risk-free term-structure plus a spread that compensates the investor for the risks. Hence whenever the risk-free term-structure, or the Swiss Franc risk changes,  $c$  has to be adapted for the value  $v(0)$  to be equal to one. This is difficult to automate, and we therefore allow the stakers to collectively set the interest rate (i.e., the risk-free rate plus spread). That is, if the exchange rate of the ZCHF is too low, the stakers use the governance mechanism to increase the interest rate and vice versa.

We design the system so that it is in the interest of the stakers to set the parameters of the system such that the peg is maintained. There should be no abuse of power, for example to set interest rates too high, therefore pushing the value of their ZCHF way beyond one CHF and essentially stealing the collateral as it would become too expensive for the minters to buy ZCHF to get their collateral back. To prevent such an attack, we ensure that stakers can only slowly adjust the interest rate, so that it is possible for the participants to trade-in their ZCHF before rates are too punitive (e.g., minters redeem their collateral by repaying their loan).

The next section presents a specific setup of mint plugins and proposes a calibration method to determine the appropriate spread that should bring the value of the Frankencoin close to a valuation of one Swiss Franc.

### 3 Specific Mint Plugin Setup

The Frankencoin system is open to accept any type of mint plugins. In this paper we describe a setup with two specific mint plugins that we consider to be of particular relevance when bootstrapping and growing the Frankencoin.

Each mint plugin  $i \in 1, \dots, K$  charges a minting fee  $f_i \geq 0$ , and can define a coupon rate paid to stakers  $c_i \geq 0$ . Depending on the plugin type, there can be additional plugin-specific parameters.

#### 3.1 Direct Peg Plugin

The simplest possible mint plugin is one that is based on a stablecoin with the same reference currency. Specifically, "off-chain" custodial stablecoins. For the Frankencoin, this could for example be the CryptoFranc (XCHF) issued by Bitcoin Suisse or the Digital Swiss Franc (DCHF) issued by Sygnum. This mint plugin allows anyone to deposit the specified stablecoin and to get Frankencoins in return. Also, the minting contract would allow anyone to convert Frankencoins back into the specific stablecoin for as long as there are any left.

Direct peg plugins have the advantage of strongly anchoring the value of the Frankencoin to one Swiss Franc by delegating the collateralization mechanism (e.g., directly backed by Swiss Francs). The disadvantage for direct peg plugins is the dependency on the issuers. Overall, direct peg are a great method to bootstrap the Frankencoin and diversify the Frankencoin system.

##### 3.1.1 Fee Calibration

In case of issuer default, stakers have to burn ZCHF equal to the amount of loss given issuer default times the exposure to that stablecoin. To compensate stakers for this risk, we charge a minting fee. Again, we have the advantage that calibration is outsourced to the market. If the value of one stablecoin trades at  $1 - \delta$  to the Swiss Franc, governance sets a minting fee equal to  $f_i = \delta$ .

### 3.1.2 Reserves

Each mint plugin defines the required reserves of ZCHF to be held against the issued volume of ZCHF. XCHF and DCHF come with their own guarantee that extend beyond the issuer default. Therefore, for the direct peg plugin and XCHF and DCHF as collateral, we require no reserves.

## 3.2 Liquid Collateral Plugin

The second type of mint plugin is designed for liquid collateral, such as Bitcoin. This type of plugin implements the use case of a Lombard loan that we motivated the paper with.

- The participants are minters, challengers, auction participants, and ZCHF stakers.
- The minter deposits collateral and thereby mints ZCHF. The ZCHF are overcollateralized at the time of minting, that is, the value of collateral deposited exceeds the value of the minted ZCHF.
- Challengers can initiate an auction process for a given position at any time. To do so, they deposit collateral of the same type. After the challenge initiation, auction participant bid for the collateral by despositing ZCHF.
  1. If, according to the auction, the value of the collateral falls below a specified threshold, the position is liquidated and the minter loses their collateral. *E.g., the collateral deposited is 1500 LUSD, 1000 ZCHF were minted. Now, the position is challenged and the best bid for 1500 LUSD closes at 1095 ZCHF. Let's assume that the threshold is 10%. Now, because  $1095 \text{ ZCHF} < 1000 (1+10\%) \text{ ZCHF}$ , the position is closed out.* The challenger earns a fee, and the bidder gets the the collateral he was bidding for. The ZCHF posted by the bidder are distributed as follows. The challenger receives a reward. An amount equal to the outstanding loan is burned:
    - If any ZCHF above the loan amount is left, the stakers get this amount
    - If the posted ZCHF are not sufficient to burn an amount equal to the outstanding loan, the stakers lose this amount
  2. If, according to the auction, the collateral value is above the threshold, the position remains in the minter's ownership. The bidder gets the challenger's collateral and the challenger get's the bidder's amount of ZCHF.

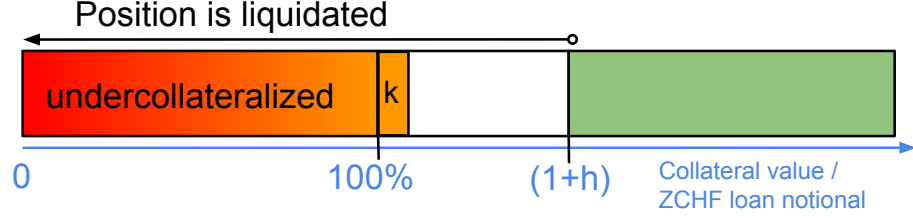
Figure 1 illustrates this auction mechanism graphically.

Our liquid collateral mint plugin has the following parameters. Variable  $h$  denotes the threshold that defines whether the position can be liquidated or not. That is, if, according to the auction, the value of the collateral is below  $Z(1 + h)$ , where  $Z$  is the amount of ZCHF minted for a given position, the position is liquidated. Variable  $k$  denotes the challenger reward (e.g., 2% of the ZCHF position). Finally,  $\tau$  is the duration of the auction (e.g., 24 hours). To summarize:

$$h : h > 0, \text{ liquidation threshold, liquidate if highest bid is below } Z(1+h) \quad (4)$$

$$\tau : \text{duration of liquidation process} \quad (5)$$

$$k : 0 < k < h, \text{ challenger reward} \quad (6)$$



**Figure 1: Auction.** Any participant can challenge a loan position consisting of the ZCHF loan and the deposited collateral. The challenger deposits collateral of the size of the loan. Auction participants bid for the collateral (here BTC) by depositing ZCHF. If the best bid price in ZCHF is above  $(1+h)$  times the notional ZCHF lent, the challenger receives the best bidder's ZCHF and the bidder the challenger's collateral (green zone). If the best bid price is below  $(1+h)$ , the borrower loses their collateral (position is liquidated). In this case the challenger gets back their collateral and earns a fee of  $k$  times the loan amount. The bidder gets the collateral deposited by the borrower. The ZCHF posted by the bidder are used to pay a reward  $k$  to the challenger,  $(1+h-k)$  times the loan amount go to the staking pool, and an amount equal to the loan amount is burned. If the ZCHF are not sufficient to burn an amount equal to the loan amount after paying  $k$ , stakers have to burn this amount of ZCHF. Hence the unshaded area in the center is where stakers earn liquidation rewards, the area to the left is costly for stakers.

Since our choice of collateral is liquid and can be traded at many different venues, we expect arbitrageurs to bid in the liquidation process and arbitrage between the liquid collateral mint plugin and other exchanges. This is best done via algorithmic trading, and we therefore propose to use a short liquidation horizon (minutes or hours, rather than days), with the lower bound being dictated by the block-time of the blockchain.

### 3.2.1 Fee Calibration

We define an "efficient liquidation process" as a situation where the collateral is challenged at a price  $(1+h)$ , and at the end of the liquidation period  $\tau$ , the collateral is filled at a competitive price.

Stakers are at risk to lose funds between the liquidation start and liquidation end. Specifically, if during the liquidation horizon the value of the collateral drops from  $(1+h)$  to below  $(1+k)$ , see Figure 1. The key challenge now is to value this risk so that the stakers are adequately compensated. To do so, we resort to the arbitrage free pricing principle, which states that the value of a contingent claim is given by its discounted expected value under the risk-neutral probability measure, see e.g., [?].

Formally, we assume a single period, denoted by  $\tau$ , a random variable,  $\tilde{r}_\tau$ , that corresponds to the log-return of the collateral over this period (using CHF as the numeraire). Now, we can express the loss to the stakers as the following random variable:

$$\tilde{L} = -\min \left[ (1+h)e^{\tilde{r}} - (1+k), h-k \right], \quad (7)$$

per unit of ZCHF minted (e.g., if the position consists of  $Z$  ZCHF, the loss to the stakers is  $Z\tilde{L}$ ). To see this, first note that the starting value of the position is  $(1+h)$  per unit of ZCHF. The value of the position at the end of the auction is  $(1+h)e^{\tilde{r}_\tau}$ . Stakers need to burn ZCHF for the amount that the end-of-auction value falls short of the minted amount plus the challenger reward, hence we subtract  $(1+k)$ . Second, the stakers earn in the liquidation process a maximal amount of

**Table 1: Minting Fees.** TODO We calibrate minting fees for a collateral with 5% daily volatility, assuming a t-distribution with 4 degrees of freedom (left), or a normal distribution (right). The volatility is scaled by  $\sqrt{\tau}$ , and the mean is assumed to be zero. Further,  $h = 0.10$ ,  $k = 0.02$ . Fees displayed are in percentages.

$\tau$	t-distr	normal
<b>1</b>	0.38	0.08
<b>2</b>	0.90	0.38
<b>3</b>	1.38	0.72

$h - k$ , see Figure 1, so we need the  $\min[\cdot]$ -function. Finally, the negative sign is convention to have a positive number for the loss.

To value the contingent claim  $\tilde{L}$ , we calculate its discounted expected value. We assume that the risk-free rate used for discounting is equal to zero, which we consider adequate especially since the period  $\tau$  is very short. Let  $f_\tau(x)$  be the density function for the return distribution over the period  $\tau$ . Now, we can value  $\tilde{L}$  as follows

$$\mathbb{E}_\tau [\tilde{L}] = - \int_{-\infty}^{\ell} [(1+h)e^x - (1+k)] f_\tau(x) dx \quad (8)$$

$$\ell = \log \frac{h-1}{h+1}, \quad (9)$$

where the subscript  $\tau$  emphasizes that the distribution depends on the time-horizon of the auction. We got rid of the  $\min$ -function of Equation (7) by integrating only up to the return  $\ell$  that yields  $\tilde{L} = h - k$ .

Equation (8) uses the risk-neutral probability measure, often referred to as  $\mathbb{Q}$ , rather than the objective measure  $\mathbb{P}$ . In practice, parameters for the measure  $\mathbb{P}$  are extracted directly from market data (e.g., sample volatilities and expected returns), whereas parameters for the measure  $\mathbb{Q}$  have to be extracted from option data under the same model assumptions (e.g., option implied volatilities). With risk averse investors, the  $\mathbb{Q}$ -measure puts more weight on adverse market events, see, e.g., [?], leading to a higher risk-neutral price of  $\tilde{L}$ , compared to the value obtained when integrating Equation (8) under the objective measure. We therefore proceed by calibrating a probability distribution to observed market data and use this as a lower bound for the price of  $\tilde{L}$ , or, equivalently the minting fee should be at least equal to the price of  $\tilde{L}$ :

$$f_i \geq \mathbb{E}_\tau [\tilde{L}]. \quad (10)$$

*TODO: BTCHF data, calibrate distribution, show results Table 1 shows results from calibrating Eq. (8). BTC-USD has a daily volatility of about 5%. The table compares the result for Eq. (8) when assuming either a t-distribution or a normal distribution for the return.*

### 3.2.2 Reserves

Each mint plugin defines the required reserves of ZCHF to be held against the issued volume of ZCHF. In the previous section we have calibrated the minimal fee that should be charged to minters via expected loss. We proceed in a similar manner and calculate the expected shortfall for the liquidity provider under a 1%-quantile, assuming all loans start at their minimal loan to value ratio at  $(1+h)$ .



## 4 Minimum Capital Requirement

In this section, we define overarching rules to make the Frankencoin system resilient. We regard the system as a bank and apply risk mitigation methods used for banks.

Figure 2 presents a balance sheet view of the Frankencoin system and to illustrate the risks, we deviate from usual accounting rules. Collateral is depicted on the asset side.<sup>1</sup> Like on a central bank balance sheet, the ZCHF in circulation are shown as a liability. The "book value of equity" is the difference between assets and liabilities. If the collateral of the liquid collateral plugin falls in value so that the book value of equity is zero or negative, the ZCHF in circulation are no longer backed and reserves have to be burned. This leads to a reduction of ZCHF in circulation and puts the balance sheet back to a healthier state.

Assets	Liabilities
ZCHF Reserves	ZCHF Staked
Direct Peg 1	ZCHF in circulation
Direct Peg 2	
Liquid Collateral Plugin	
	Book value of Equity

**Figure 2: Balance Sheet.** This diagram schematizes the balance sheet of the Frankencoin system for a specific mint plugin setup.

We motivate the following risk mitigating measures from the Basel III banking regulation and the Dodd-Frank Act, see, e.g., [?] and [?]:

1. *Risk-based capital requirements*: each mint-plugin defines its own risk-based reserve requirement. E.g., for the liquid collateral plugin, we choose to estimate a reserve requirement based on expected shortfall. The fact that the system used shared capital but individual reserve requirements adds a diversification benefit.
2. *Leverage limits* provide a supplementary measure to risk-based capital requirements due to their simplicity and independency. It is a safeguard against model risk and was introduced with the Basel III regulation. Our system is subject to model risk too, for instance due to the risk-based calibration of the Liquid Collateral Plugin capital reserve, and we thus define leverage limits too.
3. *Countercyclical buffer*: risk-based capital requirements show cyclicity. In crisis times, more capital is required than during expansionary periods. It is more difficult to attract capital in exactly these times. Therefore we introduce a countercyclical buffer, that governance reduces in downturns and increases during expansionary periods.<sup>2</sup>

<sup>1</sup>Per accounting rules, securities for collateralized credits are not reported as assets if they cannot be sold by the entity without default of the borrower.

<sup>2</sup>E.g., loss given default is typically higher in a crisis. The Basel II framework required banks to use downturn LGD estimates to dampen countercyclicity, see [?]. Basel III introduced a countercyclical buffer to dampen the cyclicity emphasized by the banking system

4. *Concentration limits.* If the collateral of Frankencoin is singularly exposed to the price of Bitcoin, the Frankencoin is doomed to collapse when the Bitcoin price falls sufficiently. We therefore aim to limit the concentration of collateral.

If the capital is below the minimal capital derived from the above measures, no more Frankencoins can be issued. The above measures provide us with a capital requirement per ZCHF issued. We propose to define the *countercyclical buffer* as a multiplicative add-on to the capital requirement resulting from the remaining measures. That is, if the requirement from 1, 2, and 4 result in a buffer of  $C$ , the countercyclical buffer is a number  $b$ , so that the minimal capital is equal to  $C(1 + b)$ . The system-wide buffer  $b$  is set by governance.

We have detailed the *risk-based capital* requirements in the previous sections. The risk-based capital requirements involve relatively complex calculations (for current blockchain capabilities), however, they are done off-chain and we are only required to store the resulting parameters in the blockchain.

A simple way to measure *concentration* is the one-firm concentration ratio, see [?], which equals the percentage of market share held by the largest firm. We apply this to our context. Let  $K$  be the number of mint plugins. Each mint plugin tracks the amount of ZCHF issued and not burned,  $Z_j$ .<sup>3</sup> The relative amount issued by plugin  $j$  is given by  $p_j = Z_j / \sum_i^K Z_i$ . We aim to prevent that the largest  $p_j$  is beyond a threshold  $\theta_C$ . If a single mint plugin reaches the threshold  $\theta_C$ , the plugin cannot issue any more ZCHF until the concentration is reduced. The concentration threshold is set by governance.

Finally, we detail the *leverage limits*. In the spirit of the creators of the leverage limits, we aim to have the least possible assumptions to define leverage limits in our system. We therefore do not want to rely on any collateral valuations or distributional assumptions. Direct Peg Plugins should not count towards the leverage ratio, because their value is stable and diversified through the concentration limits, and because they provide a convenient way to issue new ZCHF that are subsequently staked. However, we want to limit the amount of ZCHF issued through Liquid Collateral Plugins relative to the ZCHF held as a reserve in the staking pool. We therefore define the leverage ratio as

$$LR = \sum_{i \in \mathcal{C}} Z_i / S, \quad (11)$$

where  $Z_i$  is the amount of tokens issued by mint plugin  $i$ ,  $\mathcal{C}$  is the set of plugin indices that are to be included in the leverage calculation (i.e., not the Direct Peg plugins), and  $S$  is the total amount of staked ZCHF. If the leverage ratio is larger than a governance set threshold  $\theta_L$ , plugins other than the Direct Peg plugins can no longer mint ZCHF.

Table 2 summarizes the parameters and data that needs to be stored to implement these risk measures. Despite the comprehensive measures, there are only a few parameters that need to be stored.

*TODO: address how credit issuance works. Proposal: last price \* 1.8 loan amount*

---

<sup>3</sup>Direct Peg Plugins do not have the notion of a position and therefore set  $Z_j$  equal to the collateral

**Table 2: Summary of Data used in Smart Contracts.** This table lists the parameters and data that needs to be stored to implement the desired risk mitigation measures.

Element To Store	Type	Description
$Z_i$	Data	Each mint plugin keeps track of ZCHF issued by the plugin and not burned
$Z, Z^c$	Data	The governance contract keeps track of the sum of $Z_i$ and the sum over the $Z_i$ issued by leverage constraint relevant contracts
$S$	Data	Amount of ZCHF staked
$b$	Parameter	Countercyclical buffer
$\Theta_C$	Parameter	Maximal relative amount of ZCHF issued by a single mint plugin to limit concentration risk
$\mathcal{C}$	Parameter	Each mint plugin defines whether they are part of the leverage ratio calculation or not
$\Theta_L$	Parameter	Maximal leverage ratio
$h, \tau, k$	Parameter	Parameters for Liquid Collateral Plugin
$\delta$	Parameter	Parameter for the Direct Peg Plugin
$f_i$	Parameter	Each mint plugin has a minting fee
$c_i$	Parameter	Each mint plugin can define a coupon rate paid to stakers (can be zero, e.g., in case of Direct Peg Plugins)

## 5 Conclusion

TODO

## References

- [Acharya et al., 2010] Acharya, V. V., Cooley, T. F., Richardson, M. P., Walter, I., of Business, N. Y. U. S. S., and Scholes, M. (2010). *Regulating Wall Street: The Dodd-Frank Act and the new architecture of global finance*, volume 608. Wiley Hoboken, NJ.
- [Basel Committee on Banking Supervision, 2010] Basel Committee on Banking Supervision (2010). Basel iii: A global regulatory framework for more resilient banks and banking systems.
- [Björk, 2009] Björk, T. (2009). *Arbitrage theory in continuous time*. Oxford university press.
- [Breedon and Litzenberger, 1978] Breedon, D. T. and Litzenberger, R. H. (1978). Prices of state-contingent claims implicit in option prices. *Journal of business*, pages 621–651.
- [Clements, 2021] Clements, R. (2021). Built to fail: The inherent fragility of algorithmic stablecoins. *Wake Forest L. Rev. Online*, 11:131.
- [Curry and George, 1983] Curry, B. and George, K. D. (1983). Industrial concentration: a survey. *The Journal of Industrial Economics*, pages 203–255.
- [Engelmann and Rauhmeier, 2006] Engelmann, B. and Rauhmeier, R. (2006). *The Basel II risk parameters: estimation, validation, and stress testing*. Springer Science & Business Media.
- [Jarrow and Turnbull, 2000] Jarrow, R. A. and Turnbull, S. M. (2000). *Derivative securities*. South-Western Pub.
- [Jorion et al., 2010] Jorion, P. et al. (2010). *Financial Risk Manager Handbook: FRM Part I/Part II*, volume 625. John Wiley & Sons.