10 | Credit Markets

Credit market lies at the heart of finance and allows us to smooth our income streams and consumption over time. If we earn more than we consume, we can lend our money to someone else by putting it into a savings product thereby earning us some future interest payment. Someone else who earns less than he consumes can then use the capital that we deposited and put it to productive use after which he pays back the loan plus some interest.

10.1 BASIC MECHANICS

INTRODUCTION

In traditional finance, there are two main modes in which credit is made available. On the one hand, we have banks that are allowed to create money to lend it out to their customers (as discussed before). These banks will typically require some collateral (e.g., a mortgage on your house, or in the extreme case, your freedom). On the other hand, peer-to-peer lending platforms match savers and borrowers. These tend to be looser on requirements and therefore riskier. In peer-to-peer lending, all the capital needs to be available a priori. In both of these modes, the capital must be returned at predetermined periods of time together with ssome interest which can be constant (fixed), or depend on some external time-dependent factors (variable).

Decentralized credit markets offer these functions on the blockchain, but differ in that they typically require relatively large amounts of on-chain collateral to prove credit-worthiness. The reason is, of course, that borrowers tend to be anonymous and as a lender we want to avoid someone running off with the principal (the amount we lent them). Very few regulations exist and there is, therefore, no consumer protection. Interestingly, on-chain credit is typically perpetual - meaning that there is no fixed limit when a loan ends. Interest is paid out continuously and the loan can be paid back in arbitrary amounts at arbitrary times. Examples of popular credit market protocols are Compound, Aave, dydx, and AlphaHomorra.

DECENTRALIZED CREDIT MARKETS

The dynamics of decentralized credit markets are shown in Figure 1. The smart contract coordinates several actors with different incentives and use-cases.

Savings

First, a *saver* or *lender* deposits a *principal* amount to the credit market which accrues interest. Typically, she will get a *credit token* that proves her deposit. For instance, if she deposits DAI in Compound, she will get cDAI tokens of an equivalent amount back (aDAI on Aave, etc.). At a later point in time, she can withdraw the principal plus the accrued interest by sending her cDAI tokens back to the smart contract. As the cDAI tokens are not tied to any one account, a saver can alternatively choose to sell the cDAI tokens on an token market as well (see next Chapter). Savers form the lifeblood of the credit markets as they are responsible for funding the reserves from which loans can be drawn.

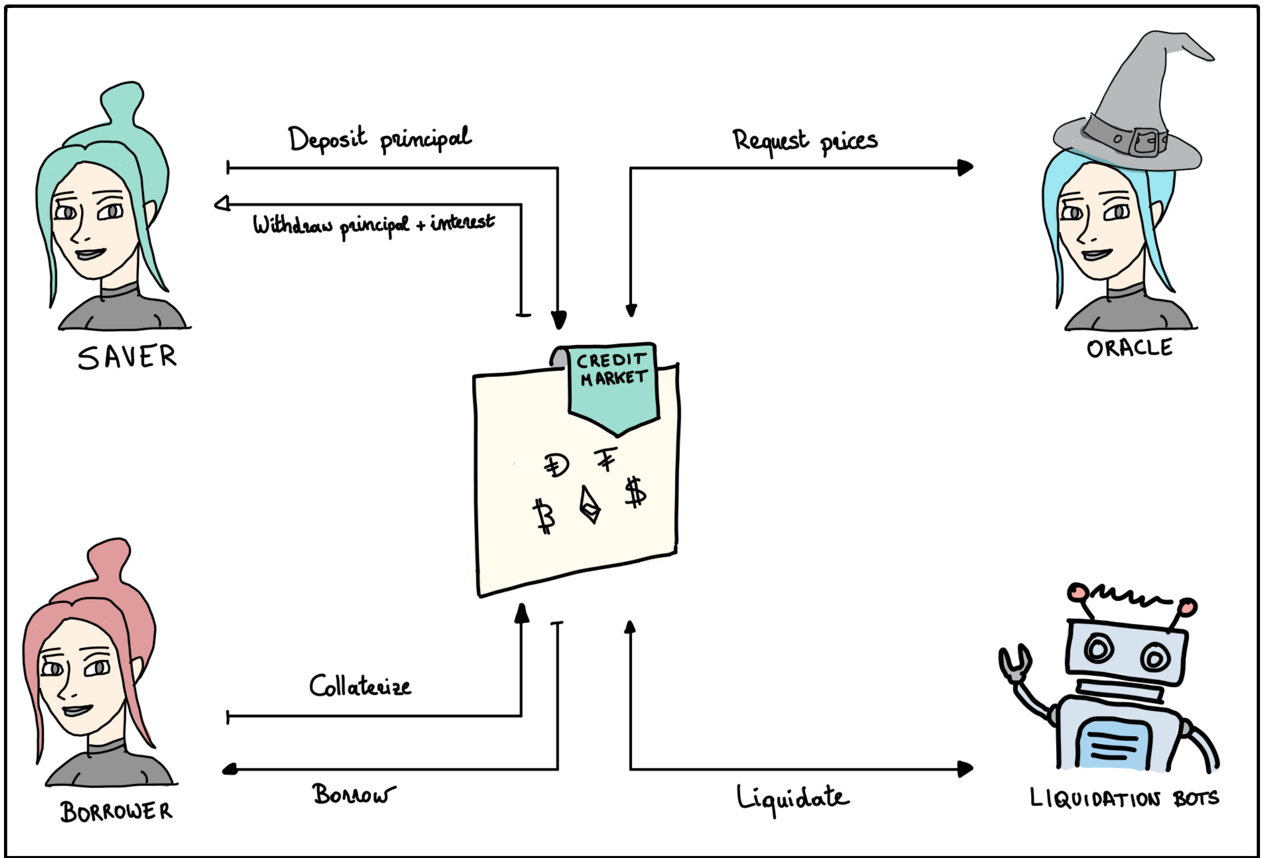

Figure 1: Decentralized credit market participants and actions.

Borrowing

A *borrower* first needs to deposit some collateral to prove that she is credit-worthy. This can be done in a similar vein as a savings deposit, the only difference being that typically only a select number of reliable tokens are typically allowed as *collateral*. After the deposit, the borrower is then allowed to borrow tokens from the credit market up to a certain amount (typically less than the value of the collateral).

Oracles

The smart contract constantly calculates the value of its assets and its liabilities based on internal and external price signals of the tokens contained in the market. The external price signals are provided by *oracles* which aggregate the price of several different trading platforms to come up with a fair estimate. This diversity of price signals is necessary as there have been cases of heavy price manipulation in single markets which have caused credit markets to fail (see the example below). Whenever a price change leads to a potential default of a borrower, the position is up for liquidation.

Liquidation bots

Liquidators (typically bots) actively monitor positions to see which ones are going in default. Upon default, they compete for the right to shore up the position by buying the debt in return for some of the collateral (and this at a discounted, profitable price).

Example: in November 2020 a price manipulation caused approximately \$89 million in liquidations, including some of the most ardent supporters of Compound. The issue was that Compound used Coinbase exclusively as a price oracle which made it open for manipulation. In this particular case, an incredibly large purchase of DAI caused its value to shoot up from its (stable) value of \$1 to \$1.30. This caused many DAI debt positions to become increasingly untenable and undercollateralized eventually going into default.

10.2 DESIGN SPACE

DeFi credit protocols differ in how they are designed. In this section, we cover the main vectors of design unique to this space and what their implications are for the usage of the respective platforms. Even though governance should be on this list, we will leave a discussion on governance for a later chapter.

COLLATERALIZATION MODEL

Terminology

The **collateral** is the sum of assets that you deposit to show your creditworthiness and to secure your debt. Platforms typically have a minimum requirement expressed as a **collateralization rate (CR)**:

$$\text{CR} = \frac{\text{Borrowed}}{\text{Deposited}}$$

We say that a position is *over-collateralized* if $\text{CR} > 1$ and *under-collateralized* if $\text{CR} < 1$. If a user's collateralization rate drops below the required ratio of a platform (e.g., 150% in some of Aave's markets), the position is (partly) liquidated. This means that other users are allowed to purchase the collateral and the debt at discounted prices. In doing so, they save you from going into complete default and they earn the *liquidation spread*. Some platforms sell off your full position, but most will only allow partial liquidations (e.g., 50% or 25% of the total position). The amount that can be liquidated in one go is called the *close factor*.

The liquidation process

In **direct liquidation** (e.g. Alpha Homorra), bots monitor the blockchain to check which positions are going in default. As soon as a position goes into default, the bot that pays the highest gas is allowed to repay the debt (at a discount) and receive collateral. This whole process is completed in one single blockchain transaction.

In **auction-based liquidation** (e.g. MakerDAO), bots bid to be able to repay the debt. Whoever places the economically most interesting bid is allowed to purchase the debt. In an *English auction*, bidders make offers as to how much they want to pay for the collateral. In a *Dutch auction*, the asking price starts high and is gradually reduced until someone accepts the price.

INTEREST RATE MODELS

Platforms must constantly balance the supply and demand of different tokens that are being deposited and borrowed. Tokens that are in high demand (but low supply) naturally should be expensive to borrow, and lucrative to deposit. Conversely, low-demand tokens with a huge supply in the credit market (e.g., ETH) should be cheap to borrow and non-lucrative to deposit. The main mechanism of regulating the cost of borrowing and saving tokens is through **interest rate models**.

The *utilization* of a market m is[1]:

$$U_m = \frac{L}{S}$$

with L loans and S savings. A high utilization rate implies that we should make the interest repayments high and vice versa. Interest rates

[1]Note that we used the word market broadly here, but in this context it often means that we analyze and set parameters for just 1 token with its unique supply & demand characteristics.

are rewarded instantaneously but only paid out when a user takes an action in the market (interacts with the credit market contract). The Interest Rate Index (IRI) at step k is:

$$I_{k,m} = I_{k-1,m}(1 + rt)$$

where t is the number of blocks that happened since the previous update (the block height) and r the per-block interest rate. The total debt is equal to the current debt, plus the outstanding interest:

$$D_{k,m} = D_{k-1,m}(1 + rt)$$

When a loan is repaid, the outstanding interest is collected by the platform and the depositors.

Kinked Interest Rates

The most common model for setting interest rates are kinked rates (Figure 2), which combine two linear curves:

$$i_b = \begin{cases} \alpha + \beta U & \text{if } U \leq U_{opt} \\ \alpha + \beta U_{opt} + \gamma(U - U_{opt}) & \text{otherwise} \end{cases}$$

where $i_b$is the borrowing interest rate, α a per-block base rate, and β a per-block multiplier based on utilization (higher utilization implies higher costs for borrowing the asset and higher interest when depositing savings in the asset).

Typically, the lenders does not get the full amount but the protocol takes a share too. In the Compound whitepaper, for example, you will find the following operator applied to their kinked interest rate:

$$i_s = U(i_b(1 - \lambda))$$

where i_s is the supply interest rate (for lenders), i_b the borrowing interest rate (for borrowers), U the utilization, and λ is a reserve factor (a percentage of the spread between i_s and i_b that the protocol keeps as profit).

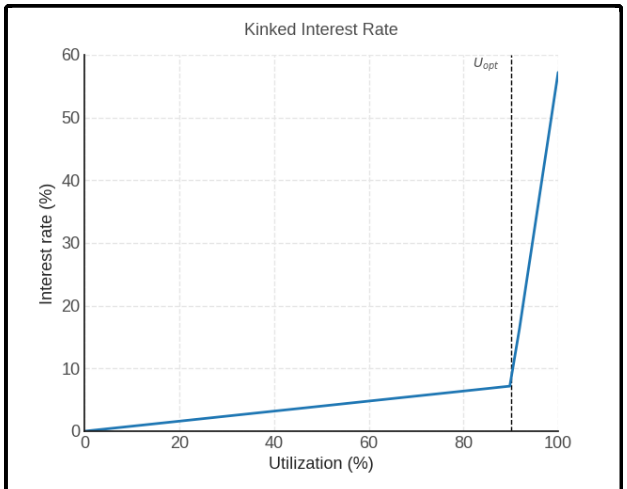

Figure 2: Kinked interest rate model

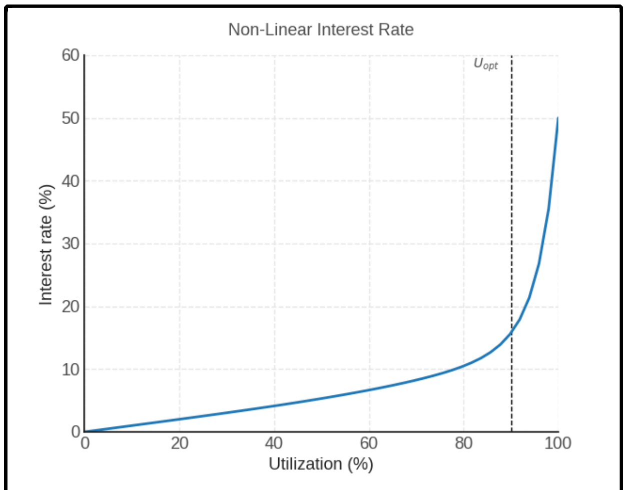

Figure 3: Non-linear interest rate model.

Non-Linear Interest Rates

The dydx model follows a non-linear curve such as:

$$i_{b,m} = 0.1U_m + 0.05U_m^4 + 0.15U_m^{16} + 0.20U_m^{32}$$

The exact components and coefficients depend on the market (asset).

CHOOSING A PLATFORM

Which platform is best depends on the use-case. Purely from an economic perspective you will want to shop around for the most profitable interest rates and lowest profit-taking platforms. In doing so, try to look at historical rates as interest rates can differ on a day-to-day basis or on weekly basis. Go for the highest when saving, and the lowest when borrowing. As some platforms are more rigid than others, they can become more interesting in adverse market conditions. As always, with crypto being so volatile, you will want to go for more established battle-tested protocols with a healthy liquidator-bot population whenever possible.

10.3 FLASH LOANS

WHAT IT IS

Flash loans are an extremely exciting technology in which anyone is allowed to (temporarily) loan astronomical amounts of assets as long as they manage to pay it back within the same transaction (using a smart contract). The process is simple:

1. Take out a flash loan
2. Go wild
3. Repay loan collateral + interest

Fees tend to be relatively high, but the amount that you are allowed to borrow is unlimited. At the time of writing these are the fees of popular protocols:

Protocol	Fee	Volume since inception
Uniswap	0.3%	8 billion USD
Aave	0.3%	10 billion USD
dydx	1 Wei	100 million USD

HOW TO EXECUTE A FLASH LOAN

Flash loans are a more advanced use-case as they require knowledge of Solidity. You write a smart-contract as in Algorithm 10.1 and then execute the code by sending a transaction to appropriate function call in the contract. This is starting to change with "nocode" -platforms (e.g., `http://equalizer.finance`) providing web applications that allow you to execute flash loans on popular protocols.

USE-CASES

Arbitrage

Flash loans grant borrowers access to great amounts of capital to execute arbitrages thereby making the market more efficient. Figure 4 shows an example of an arbitrage situation that was executed on July 31st, 2020 in transaction 0xf749..[2] on the Ethereum blockchain using dydx. In this particular arbitrage, the arbitrageur exploited the fact that the price of USDC/DAI is different on two different exchanges of the curve protocol. This mispricing allowed the arbitrageur to bridge the market as follows:

1. Take out a loan of 2.048M USDC
2. Exchange the 2.048M USDC for 2.028M DAI in market 1 (at price 1.010 USDC/-DAI).
3. Exchange the 2.028M DAI back to 2.064 M USDC in market 2 (at price 1.018 USDC/-DAI).
4. Repay the loan 2.028M USDC loan.
5. Profit 16,182 USDC

The flash loan has given the arbitrageur a juicy profit without having to deposit any funds! Participants of the Curve market are not necessarily unhappy as their market has now become more efficient with the USDC/DAI price being more equal after the arbitrage.

[2]0xf7498a2546c3d70f49d83a2a5476fd9dcb6518100b2a731294d0d7b9f7

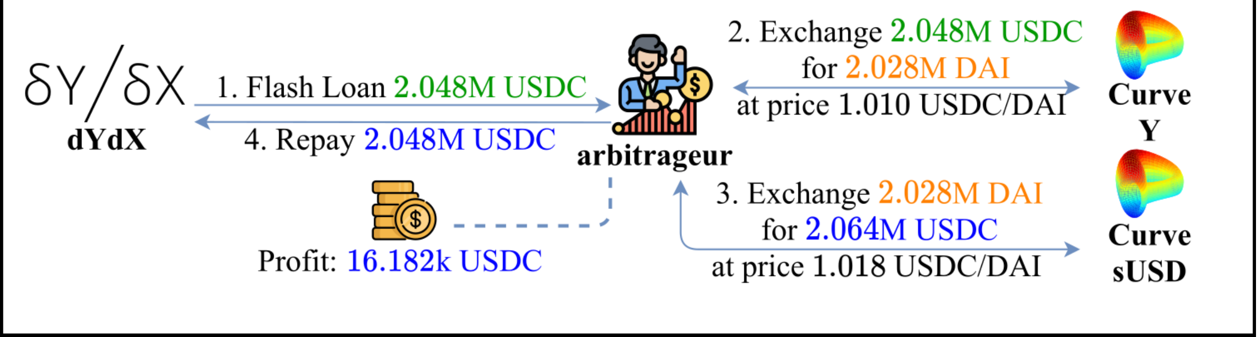
δY/δX
dYdX
1. Flash Loan 2.048M USDC
4. Repay 2.048M USDC
arbitrageur
2. Exchange 2.048M USDC for 2.028M DAI at price 1.010 USDC/DAI
Curve Y
3. Exchange 2.028M DAI for 2.064M USDC at price 1.018 USDC/DAI
Curve sUSD
Profit: 16.182k USDC

Figure 4: Arbitrage example using a flash loan (Source: Qin et al 2021)

DeFi attacks

Flash loans are often used in DeFi attacks. The attacker borrows a massive amount of tokens to then manipulate the price on a market which is used in some other platform as a price oracle (as was the case in the previous example on Compound). This is the DeFi equivalent of a pump & dump and can be easily resolved by using better price signals such as ChainLink oracle technology. In further reading, we provide resources to more in-depth examples of this type of attack.

Washtrading

Washtrading is the act of increasing the trading volume of a market by repeatedly buying/selling assets on that market. Volumes are often used as a signal for popularity and potentially future price in DeFi markets. Furthermore, in TradFi, washtrading has been used to generate commission fees that are redirected to fraudulent administrators who can hide these income streams better (like in the Libor scandal in traditional finance).

Flash-liquidation

Furthermore, they enable *flash-liquidations*, where the liquidator does not have the needed liquidity at the beginning of the trade to execute a liquidation but instead borrows it and then repays the debt using the received collateral.

Collateral swapping

Lastly, they enable *collateral swapping* to efficiently swap a debt position.

Example: collateral swapping

Assume that you have a debt position with 1 ETH collateral deposited and a 1000 DAI liability. You decided that you want to swap your ETH to USDC but want to keep your DAI in productive use.

1. Take a flash loan of 1000 DAI
2. Repay 1000 DAI to close your current debt position.
3. Redeem 1 ETH.
4. Swap 1 ETH to 2000 USDC
5. Collateralize 2000 USDC and open a new position where you borrow 1000 DAI.
6. Repay the flash loan.

Algorithm 10.1 Example of a flash loan on Aave v1.0

```
pragma solidity ^0.6.6;

import "FlashLoanReceiverBase.sol";
import "ILendingPoolAddressesProvider.sol";
import "ILendingPool.sol";

contract Flashloan is FlashLoanReceiverBase {
    constructor(address _addressProvider) FlashLoanReceiverBase(
   _addressProvider) public {}
    /**
        This function is called after your contract has received the flash
   loaned amount
     */
    function executeOperation(
        address _reserve,
        uint256 _amount,
        uint256 _fee,
        bytes calldata _params
    )
        external
        override
    {
        require(_amount <= getBalanceInternal(address(this), _reserve),
                "Invalid balance, was the flashLoan successful?");

        // Go wild.
        // !! Ensure that *this contract* has enough of `_reserve` funds to
   payback the `_fee` !!

        uint totalDebt = _amount.add(_fee);
        transferFundsBackToPoolInternal(_reserve, totalDebt);
    }
    /**
        Flash loan 1000000000000000000 wei (1 ether) worth of `_asset`
     */
    function flashloan(address _asset) public onlyOwner {
        bytes memory data = "";
        uint amount = 1 ether;

        ILendingPool lendingPool = ILendingPool(addressesProvider.
   getLendingPool());
        lendingPool.flashLoan(address(this), _asset, amount, data);
    }
}
```

10.4 RESOURCES

- "An Empirical Study of DeFi Liquidations: Incentives, Risks, and Instabilities", Qin et al 2021
- "DeFi Protocols for Loanable Funds: Interest Rates, Liquidity and Market Efficiency", Gudgeon et al 2020.
- Aave Protocol Whitepaper v1.0
- "Compound: The Money Market Protocol", Leshner, Hayes 2019
- "dYdX: A Standard for Decentralized Margin Trading and Derivatives" Juliano 2017