Angle Borrowing Module Whitepaper

Angle Labs Core Team

contact@angle.money

June 1, 2022

Abstract

We propose here a new over-collateralized minting module for Angle Protocol to let people borrow the protocol's stablecoins (only `agEUR` so far) while keeping their exposure to their collateral. This module would be used in parallel to Angle's existing module, with both modules having a minting right on `agEUR`.
It could be safely deployed on multiple chains and serve as a basis for Angle to be backed by highly volatile assets or to create stablecoins pegged to other assets than the Euro without having to care about finding demand for perpetual futures of the protocol.
This module comes with several improvements over the more traditional vault-based designs for stablecoins implemented by Maker [1], Liquity [2] or Abracadabra [3].

Contents

1 Introduction

Angle Protocol [4] has successfully launched in November 3rd 2021 its first over-collateralized and capital-efficient module. agEUR, the only stablecoin of the protocol so far, has kept a robust peg.

While the protocol has remained over-collateralized thanks to its numerous Standard Liquidity Providers, it has not been fully hedged all the time by its HAs. Reasons for not being fully hedged may be due to the high fees for Forex with respect to the volatility of the underlying, to the lack of DeFi Forex traders or to the difficulty to understand the product.

On top of that, agEUR has mostly been used for farming purposes, and its growth has been slowed by the fact that some people are not ready to completely give up on their collateral to get the stablecoin.

Overall, we believe that demand for agEUR as well as the robustness of Angle could be boosted by having a traditional over-collateralized vault-based minting module.

This conventional stablecoin design, even if it has several drawbacks (capital inefficiency to mint stablecoins, no direct arbitrage for peg) has multiple advantages compared to Angle current one:

- It has made its proofs over a long time period in terms of robustness and can pretty well resist black swan events with volatile collateral types as collateral.
- It is easily replicable on other chains: the only thing needed to make it work is good liquidators and easily accessible liquidity for liquidations. With Angle's current model, bootstrapping the protocol on a chain requires carefully growing the joint demand for stablecoins, SLPs and HAs. This makes it risky to deploy the protocol in multiple places at once.
- People minting stablecoins from such modules are still exposed to the volatility of their collateral: when depositing ETH and borrowing agEUR against it, you still own your ETH as collateral and are exposed to the upside or downside of it.
- If well thought, such designs can also technically allow users to get leveraged in just one transaction.

In order to bring robustness to the system and to expand use cases around agEUR, we are therefore proposing to add an over-collateralized vault-based module to Angle. Both would work as complements of each other (in particular for arbitrageurs).

In this paper, we present the main aspects of the design of what we'll often refer to as Angle Borrowing Module.

2 Prerequisites

To understand Angle Borrowing Module design, it is essential to have some essential prerequisite concepts in mind first:

- **Vault-Based Over-Collateralized Stablecoin Protocols**: It refers to stablecoin protocols in which stablecoins can be borrowed by users out of collateral. Collateral can be redeemed upon repayment of a portion or all the stablecoins borrowed plus interest. These protocols are called over-collateralized because usually there is more collateral in value than stablecoins issued. They often have different peg mechanisms built-in to make the stablecoins really stable. They're very similar in some way to lending protocols.
- **Vault**: Also called Position. It's a core component of vault-based modules which facilitates the generation of stablecoins against locked up collateral. A vault is defined by the amount of collateral in it, and by its debt (see below).
- **Debt**: It corresponds to the amount of stablecoins borrowed in a vault compounded by the interest rate. The debt of a vault which borrowed 100 stablecoins at a 1% interest rate will be 101 after a year.
- **Collateral Ratio**: Ratio between the amount of collateral in a vault expressed in stablecoin value and the debt of the vault.
- **Collateral Factor**: Factor used to reflect the risk of an asset that's being used as collateral in the protocol and to compute the minimum collateral ratio possible for a vault. The riskier an asset (the more volatile or illiquid it is) the smaller the collateral factor and hence the bigger the minimum collateral factor should be.
- **Unhealthy Vault**: Vault with a collateral ratio below the minimum collateral ratio allowed by the protocol.
- **Liquidation**: Process by which all or a portion of the debt of an unhealthy vault is repaid against collateral from this vault given at a discount to external incentivized actors.

Like all vault-based stablecoin protocols, Angle Borrowing Module allows users to borrow stablecoins against collateral placed in excess value, to repay just a portion of their debt, to add more collateral, or to remove some collateral at any time. While all of this is relatively common, we are introducing below the main and key distinguishing features of what could become a new module for Angle.

3 Base Features

3.1 Multi-Collateral

This vault-based over-collateralized module supports different assets as a collateral, and for a same collateral there can be different parameters against which users could borrow depending on the risk-level of the collateral. This is in a way similar to Maker which accepts `ETH` as a collateral with different versions of risk parameters.

EXAMPLE 3.1

The new module could let you borrow `agEUR` from `ETH`:

- With a 2% yearly interest and a minimum collateral ratio of 180%
- Or with a 4% yearly interest, a 0.5% minting fee but a minimum collateral ratio of 150%

Basically, the smaller the collateral ratio (or the higher the collateral factor) for a given asset, the riskier the asset is and the more fees should be taken by the protocol.

3.2 Isolated Positions

Positions (or vaults) across different collateral types are isolated. This is similar to what is done in other stablecoin protocols.

EXAMPLE 3.2

> Someone can borrow `agEUR` from `ETH` and from `wBTC` but both borrowing situations are isolated from one another.

In addition, and this is not the case everywhere, an address can own several independent vaults for the same collateral type.

EXAMPLE 3.3

> Someone could also decide to open with the same address two independent positions from `wBTC` one with a higher collateral ratio than the other. The consequence is that one could get liquidated while the other not.

What allows this composability is the fact that each position is encoded as a NFT. Vaults therefore enjoy the same properties as any `ERC-721` token: they can for instance be transferred, or approval can be given by an address on a specific token.

There will be as many NFT contracts as there are different collateral/parameter types within the protocol. For instance if we were in Maker case, `ETH-A` and `ETH-B` would be two different NFT contracts.

We call these NFT contracts `VaultManager` contracts.

3.3 Dust

To make liquidations profitable in all cases for liquidators and hence the protocol viable, each position must keep a minimum of debt at all times. This is encoded by a parameter called `dust`. This is going to prevent small wallets from opening too small `agEUR` positions.

3.4 Fees

Two types of fees can be taken by this module:

- Interest rates (also called "stability fees"): `agEUR` minted from this module are borrowed and should be repaid with an interest. The amount of interest paid depends on a rate controlled by governance.
- Upfront minting fees

  EXAMPLE 3.4

  > You want to mint 100 `agEUR`, and there is a 1% fee: the person would end up getting 99 `agEUR` and 1 would be taken as a fee by the protocol.

The fact that the code allows for such fees does not mean that they will necessarily end up being taken in the end. Governance could very likely set 0 upfront fees.

Stability fees serve different means:

- Accumulate reserves for riskier assets in case many positions get simultaneously under-collateralized and lead to the creation of bad debt where less collateral backs the stablecoins issued.

- Maintain peg of agEUR in extreme market conditions. While the peg should be easily maintained thanks to agEUR deep Curve liquidity and to Angle's current system which allows direct arbitrages, increasing the interest rate for agEUR could lead people to repay their debt by burning their stablecoins and hence to increase the price of agEUR. The opposite could happen after a decrease of the stability fee.
- Accumulate surplus for the Angle ecosystem. This is one of the main lever to provide value to veANGLE holders as a portion of these stability fees could be distributed to them. Contrarily to what is done in other stablecoin protocols like Abracadabra, collateral in vaults is not invested, which means that the protocol will not be making any revenue from its TVL per se.

In summary, each collateral type whitelisted to allow borrowing of agEUR against it is associated to several key parameters:

- Stability fee
- Minting fee
- Minimum collateral ratio (encoded through the collateral Factor)
- Target health factor: more details below, but this parameter gives the objective for liquidators to liquidate to.

4 Liquidations

4.1 Background

The robustness of an over-collateralized vault-based system is bound to how efficiently unhealthy vaults can be liquidated. A flawed liquidation mechanism could leave the system in a state where the value of the collateral backing the stablecoins is inferior to the value of the stablecoins.

Angle Borrowing Module's liquidation mechanism is based off Euler's one [5] where liquidators can repay someone's outstanding debt in just one transaction without having auctions, and get a portion (or all) of the collateral of the position at a discount.

4.2 Liquidation Surcharge

The protocol takes a small fee on each liquidation to accumulate reserves for assets which tend to be often liquidated and could lead to the creation of bad debt. Fees are paid in stablecoins by liquidators which are then reimbursed in collateral by the liquidated vault.

4.3 Variable Liquidation Amounts

Contrarily to Aave V2 and Compound [6] where liquidators can only repay 50% of a debt during a liquidation, but like in Aave V3 [7] and in Euler, the amount to repay during a liquidation is dynamic and computed such that after a liquidation, the position liquidated ends up in a level of "health" defined by a target parameter: the target health factor.

In some conditions, liquidators may be able to liquidate all of a position's available debt or collateral to avoid leaving an amount of debt that is too small for the owner to be incentivized to come to repay.

In most cases though, liquidated vaults may get less than 50% of their position liquidated: this system is designed to be **far more borrower friendly than existing alternatives.**

Mathematically speaking, the health factor of a position is:

$$HF = \frac{c \times CF}{d}$$

Where:

- c: Collateral amount in the vault (expressed in stablecoin value)
- d: Debt of the position
- CF: Collateral factor of the asset

EXAMPLE 4.1

If Alice owns vault with 10 ETH and if 1 ETH is worth 2000€, then $c = 2000 \times 10 = 20000$.
If $CF = 0.75$, and Alice borrows 2000€, then: $HF = \frac{20000 \times 0.75}{2000} = 7.5$

A position can be liquidated when its health factor becomes inferior to 1. As such, a collateral factor of $\frac{2}{3}$ means that the minimum collateral ratio of a position is 150%.

If a liquidator repays x of stablecoins to the protocol and the liquidation surcharge is s and the discount proposed to the liquidator is e, then the health factor post liquidation is:

$$HF_{post}(x) = \frac{(c - \frac{x}{1-e}) \times CF}{d - x(1-s)}$$

Note that HF_{post} is not always an increasing function of x. It is the case iff at the start of the liquidation:

$$HF = \frac{c \times CF}{d} \geq \frac{CF}{(1-e)(1-s)}$$

While overall the protocol should aim for liquidations to happen in areas where liquidating increases the health factor, there are still two cases to distinguish:

- If the health factor at the start of a liquidation is greater than the term on the right, then liquidating increases the health factor and liquidators can reimburse a maximum amount x_{max} such that:

$$HF_{post}(x_{max}) = \texttt{target health factor}$$

- If the health factor is inferior to the term on the right, then liquidating necessarily decreases the health factor of the position. In this case, the liquidator can liquidate everything that's in the position. Note that amount repaid is capped here because a liquidator can never get more collateral than what was initially in the position.

$$x_{max} = c \times (1-e) \leq \frac{d}{1-s}$$

Bottom line is that the maximum amount repaid by liquidators during a liquidation depends on specific conditions of the health factor, and is therefore not fixed (like in Aave V2 or Compound) to 50% of the debt of the liquidated position.

4.4 Dust Amounts Handling

As explained above, too small (dusty) debt positions are prohibited within the new module to always keep profitable opportunities for liquidators to come liquidate unhealthy vaults.

As such following the different cases above:

- If the health factor is an increasing function of x: the protocol makes sure that the liquidator does not leave a dusty amount in the vault after repaying a portion of the debt. In some situations, it could hence allow liquidators to liquidate the whole position that is to say to repay x such that:

$$x = \frac{d}{1-s}$$

Leftover collateral in the vault would be in this case in stablecoin value:

$$c - \frac{d}{(1-s)(1-e)} \geq 0$$

- If liquidating decreases the health factor, a similar parameter is introduced to make sure that after liquidation a too small amount of collateral does not stay in a vault. Liquidators could be forced to liquidate everything and make sure there is no collateral left in a position. In these situations, leftover debt in the vault would be:

$$d - c(1-e)(1-s) \geq 0$$

This leftover debt is called bad debt as it corresponds to stablecoins that are not backed by any collateral.

4.5 Bad Debt Tracking

As full liquidations can be enforced, the protocol can safely track its bad debt, that is to say the amount of stablecoins in circulation which are not backed by anything. Bad debt is only recorded when a vault is fully liquidated with no collateral left inside.

Bad debt (just like surplus) is going to be pooled across all `VaultManager` contracts to make sure that no surplus is distributed to `veANGLE` holders if bad debt accrued somewhere for one collateral type.

4.6 Dynamic Discount

Like in Euler, the discount proposed to liquidators is variable. While liquidations usually happen on a first come first served basis, because discount is dynamically computed, here we expect that the liquidator which can liquidate most efficiently and at the smaller price for them will do.

The discount factor for a position grows linearly as the position gets away from its first possible liquidation point. It creates Dutch auctions for liquidators and ensures that the protocol gives to liquidators just what is acceptable to them and that liquidated vaults do not lose too much. Liquidation experience should hence be more borrower friendly than other stablecoin protocols.

Mathematically speaking, the value e of the discount is a function of the health factor and writes:

$$e(HF) = a(1 - HF)$$

Where a is a slope parameter referred to as the liquidation booster. This discount is capped by a maximum discount amount.

One disadvantage of this design is that if a position hits its liquidation threshold $HF = 1$, liquidators may not directly jump in because the discount proposed is too low for it to be profitable. Governance will thus have to set the collateral factor and discount parameters with this mind.

EXAMPLE 4.2

> If the minimum collateral ratio is 150%, depending on the value of a, vaults may only be liquidated when

collateral ratio reaches 140%. At 150%, discount proposed is indeed 0.

4.7 Liquidation Boosters

The liquidation mechanism is designed such that the protocol could technically favor liquidators which are already stakeholders of the protocol. The booster parameter a can indeed be made different depending on the liquidator.

In a first implementation, a is going to be a function of the `veANGLE` adjusted balance of a liquidator. Liquidators are expected to be smart contracts that are not whitelisted to own `veANGLE`, and a `veANGLE` delegation feature to delegate boosts to some contracts can be easily launched (similar to what Curve has already done).

In this scenario:

$$a = \text{baseBoost} \times f(\texttt{veANGLE Balance})$$

Where f is a piecewise linear increasing function of the `veANGLE` balance capped by a certain amount.

4.8 Conditions on Discount

When setting parameters, governance should make sure that when a health factor enters in its area where it is a decreasing function of the repay amount, the discount proposed to liquidators should be equal to the maximum discount. Otherwise it could profitable for liquidators to liquidate in multiple times as it would decrease the health factor and therefore increase the discount.

We always have:

$$\frac{CF}{(1-e_{max})(1-s)} \geq \frac{CF}{(1-e(HF))(1-s)}$$

As such, if the increasing health factor condition ($HF \geq \frac{CF}{(1-e)(1-s)}$) is verified for e_{max}, then it is necessarily verified for any lower e.

Defining e_{max} and s gives a lower bound on the health factor for which it can increase after a liquidation.

EXAMPLE 4.3

$e_{max} = 0.2, s = 0.02, CF = \frac{2}{3} \implies HF = 0.85$. From this a lower bound on the value for a can be found: $a(1-0.85) = 0.2 \implies a = \frac{4}{3}$

4.9 Minimum Capital Requirements

The module is designed such that liquidators can liquidate without any capital requirement: in the liquidation process, discounted collateral is first given to liquidators for them to find the `agEUR` they need to liquidate the position (like done by Maker in their `clip` mechanism). Flash loans are not even be needed in all cases to liquidate: this makes liquidations truly open for the Angle Protocol.

On top of that, liquidators can liquidate different positions within a similar `VaultManager` contract in just a single transaction involving only one stablecoin and collateral transfer thus making it more capital and gas efficient for them.

Overall, we expect that people will liquidate using `agEUR` from different sources:

- The module that is already live: people could mint agEUR using USDC, DAI, FEI or FRAX to participate in liquidations. For these specific collateral types, discount should not go over the minting fees on the current module and an epsilonesque amount (neglecting gas cost), as the liquidator can mint at oracle value and without slippage.
- Treasury built by liquidators alone or by pools to make liquidations more efficient for them and share fees together
- Uniswap/DEX liquidity on which liquidators may face a slippage when swapping collateral for stablecoins.

Ultimately, this liquidation design was thought to be borrower friendly while still allowing for efficient and permissionless liquidations.

It leaves many parameters up to the governance and requires this governance to be wary when setting parameters to make sure liquidations could be profitable for the protocol, for liquidators and not too harmful to borrowers in all cases.

5 Advanced Features

5.1 Debt Transfer

As explained above, different collateral types are segregated with one and other with this new module.

EXAMPLE 5.1

> If you deposit 150€ worth of ETH and 150€ worth of DAI, you will not be allowed to borrow 200 agEUR from this: your ETH gives you the right to a borrow amount, and you can also borrow agEUR out of your DAI, but not of the combination of the two positions.

One of the innovations proposed here is the ability to transfer debt between two positions you control. If you control an ETH and a DAI vault, you can transfer the debt associated to one vault to another one to balance your risk and the borrowing power associated to the two.

EXAMPLE 5.2

> If I have one position with as collateral 100 € worth of ETH and 50€ borrowed, and a DAI position with 100€ worth of DAI and 1€ borrowed: I can transfer my 50€ debt in ETH to my DAI position which now still has 100€ as collateral but a borrow amount of 51€. In this situation, my ETH position is no longer close to be liquidated as the borrow amount is now null.

Debt transfer is just an accounting procedure that eliminates the need to actually transfer agEUR between one position to another and greatly simplifies vault management.

5.2 Capital-Efficiency and One Transaction Leverage

Users willing to interact with their position can undertake many different types of actions with their vaults: create a vault, close it, add collateral, remove collateral, borrow stablecoins, repay stablecoins, get debt in their vault from another vault.

In Angle Borrowing module, all these actions are composable with one another meaning that they can all be done in any order and in just one transaction. Besides the obvious gas efficiency of this, other advantages are that it increases capital efficiency for vault interaction and enables some features like one transaction auto-leverage.

For instance, in the case of someone willing to add collateral to borrow stablecoins, then the contract will first give this address the borrowed stablecoins before asking for the collateral to be added: this means the user could swap a portion of the obtained stablecoins for collateral in the same transaction.

Mathematically speaking, someone with an amount c of collateral can borrow at most $c \times CF$ of stablecoins (where CF is the collateral factor), which when swapped to collateral and brought back in the position can be used to borrow $c \times CF^2$ of stablecoins and so on.
With an initial amount c of collateral, an address can end up if it can swap the obtained stablecoin with zero fees with an exposure to the collateral equal to:

$$\sum_{i=0}^{+\infty} c(CF)^i = \frac{c}{1 - CF}$$

With $CF = \frac{2}{3}$, you can get a 3x leverage in just one transaction.

In a similar fashion, people looking to close their vaults will be able to do so by first getting in the transaction the collateral from their position and then by repaying the stablecoins they owe which makes it more efficient for them: they can just swap the collateral of their position to stablecoins.

5.3 Flash-Loan Feature

This is separated from the module here, but in parallel to it, a new feature proposed is a flash loan feature for `agEUR` (like what `DAI` has).

`agEUR` given out in flash loans are freshly minted for the need of the flash loan and are burnt from the contract at the end of the flash loan. Uncapped and zero fees flash loans on `agEUR` could thus be available if governance chooses to set parameters that way.

6 Reactor

6.1 Whitelisting and Volatile Assets as Collateral

Since the module is designed to have only one contract per collateral type to handle positions, some specific collateral types can easily be made permissioned meaning that only some addresses will be allowed to borrow out of it.

This could serve different use cases:

- For specific collateral types (like real-world assets used as collateral): governance may need to keep tight control on who can borrow to avoid exploits.
- For some DAOs: their token could be used as a collateral to borrow `agEUR`. In this situation, to increase the safety of the protocol, only a specific Angle contract could own and control the position to make sure the borrowed `agEUR` are lent in some whitelisted destinations (most likely yield bearing strategies). This brings us to another feature of the new module which is Angle Reactor Mode.

6.2 Reactor Mode and Strategies

For `ANGLE` at first, and maybe other riskier assets, the module can be used in **Reactor Mode**. In this mode, users can deposit collateral, and the protocol opens debt and manages the `agEUR` borrowed on their behalf, investing it in yield strategies. The protocol then splits the obtained yield between itself and the depositors.

Yield strategies for agEUR would first include Curve/Convex single-sided deposits as well as Aave, Euler and Rari lending.
The agEUR (or stablecoins) minted from reactors bare a minimal risk: indeed, in case of a price drop of the collateral backing it (ANGLE price drop for instance), as it's the protocol that controls the strategies and hence the agEUR, it can pull the funds from the strategies and burn it before any possible liquidation. Liquidations could still happen though in case strategies experiment a loss.

Upside is that it brings yield opportunities for ANGLE holders and revenue for the protocol.

This is in fact close to what FEI is proposing with Turbo, and it opens the gate to DAOs making efficient use of the tokens they have in their treasury to get some yield. Implementation of reactors by the way follows EIP-4626.

It is also close to what FRAX is doing with its AMOs, except that the FRAX invested in AMOs are not backed by anything (and a black swan event or a strategy loss could create a huge bad debt for the FRAX protocol), while the agTokens (in the beginning agEUR) issued from the reactors are backed by ANGLE (or another token).

7 Settlement Mechanism

7.1 Rationale

Like in any over-collateralized stablecoin protocol, the settlement mechanism is essential to handle the case where the protocol has a too important bad debt.
Each VaultManager can be settled independently of the other, meaning this settlement procedure could be used to remove a risky asset from the protocol. A global emergency shutdown of all VaultManager contracts could also be used to end the module overall.

7.2 Design

Like in Maker's case, the settlement of a VaultManager contract works in several steps:

1. Oracle value and all operations are frozen, collateral available in the VaultManager is sent to the settlement contract.

2. Owners of over-collateralized vault have several days during which they can claim the collateral in their vaults by bringing back all the stablecoins they owe. This first step ensures that owners of over-collateralized vaults remain unaffected if the protocol gets in a loss scenario or is under-collateralized as a whole.

3. After some time, owners of stablecoins can get leftover collateral at oracle value. This is the only place within the module where stablecoins can be exchanged at oracle value against collateral. If the amount of collateral left is less than the outstanding value of the stablecoins, then stablecoin holders do not receive an equal amount of collateral.

4. Any excess collateral will be available for governance to use.

8 Implementation Details

8.1 Access Control

Access control for the whole module is centralized in just one single contract responsible for handling all role-related aspects. This reduces storage cost across all contracts.

8.2 Debt Ceiling

To better control the risk associated to each asset used as a collateral, governance includes like in Maker and in Aave V3 a total borrow cap per asset. Thanks to this, the protocol can efficiently balance the risk associated to the different collateral types used.

8.3 Oracles

Different oracle contracts than the ones used in the first module need to be used. The protocol can now afford to use only a single Chainlink [8] feed to price its collateral types (for instance `ETH`) in Euro.

8.4 Debt Tracking and Computing Interest Rates

Tracking the debt of a position is something that is not obvious when there are varying interest rates. To do this efficiently, this new module stores in each vault an internal value of the debt and across all vaults a global variable called the interest rate accumulator (or IR).

If the interest rate per second paid by borrowers is r and if $\Delta_t = t_1 - t_0$ seconds have elapsed since the last update of the interest rate accumulator, then:

$$IR_{t_1} = IR_{t_0} \times (1 + r)^{\Delta_t}$$

If someone borrows an amount y of stablecoins at some point, then what is stored in terms of debt for this person is `Internal Debt` $= \frac{y}{IR}$.

The interest accrued by the protocol between two timestamps is, if the debt has not changed:
`Total Internal Debt` $\times (IR_{t_1} - IR_{t_0})$.

The value of the interest rate accumulator does not have to be modified at each timestamp if the interest rate value or the total debt do not change. The reason for this is that if you know how much time has elapsed since the last call you can easily recompute the correct value.

For instance between t_0 and t_1: $IR_{t_1} = IR_{t_0} \times (1 + r)^{t_1 - t_0}$. And between t_0 and t_2: $IR_{t_2} = IR_{t_0} \times (1 + r)^{t_2 - t_0}$.

But this is equal to:

$$IR_{t_2} = IR_{t_0} \times (1 + r)^{t_2 - t_1 + t_1 - t_0} = IR_{t_1}(1 + r)^{t_2 - t_1}$$

Angle Borrowing Module, by minimizing variables in storage as well as storage updates is thus more gas efficient than Aave, Compound and Euler to track accruing debt.

8.5 Interest Rate Computing

There is nothing obvious with power computation in Solidity and it can rapidly lead to overflows. As such compounding interest rates (like in the formula above) is not that easy.

While Maker uses its own library for interest rates, the solution adopted here, like Aave in their V2 and V3 is to rely on a third order binomial approximation which works assuming a small interest rate.

$$(1 + r)^{\Delta_t} \approx 1 + \Delta_t r + \frac{1}{2}\Delta_t(\Delta_t - 1)r^2 + \frac{1}{6}\Delta_t(\Delta_t - 1)(\Delta_t - 2)r^3$$

This approximation shows up to be quite accurate even for big time frames (like 10 years). It however slightly undercharges borrowers with the advantage of a great gas cost reduction. Interest rates could be set by the protocol with this in mind.

9 Perspectives

While this new module follows a traditional over-collateralized vault-based design, it is designed to make the best of existing approaches, to incorporate some ideas from the most efficient lending protocols and to introduce some innovations.

Debt transfer, capital efficient liquidations, improved borrowing and more generally vault management experience, gas efficient operations are some of the key features here which make Angle design cutting-edge.

When launched, this module will work in conjunction with the one already in production. With this, Angle could be seen as:

- An innovative derivatives-backed stablecoin protocol with a built-in borrowing module
- An improvement over Maker, Abracadabra, Liquity, or even Compound, Euler and Aave but with a price stability module under steroids.

Governance will have all the tools in hands to balance risks between the different modules and make sure that `agEUR` and Angle future stablecoins remain decentralized, safe-to-use and over-collateralized stablecoins.

References

[1] The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System. https://makerdao.com/en/whitepaper/#in-mcd-we-trust/. [Online; accessed 8-March-2022].

[2] Liquity: Interest Free Borrowing on Ethereum. https://docs.liquity.org. [Online; accessed 8-March-2022].

[3] Abracadabra: Protocol Documentation. https://abracadabra.money. [Online; accessed 8-March-2022].

[4] Angle: A Decentralized and Over-Collateralized Stablecoin Protocol. https://docs.angle.money/whitepaper. [Online; accessed 8-March-2022].

[5] Euler Finance: Permissionless Crypto Lending Markets. https://docs.euler.finance/getting-started/white-paper. [Online; accessed 8-March-2022].

[6] Compound: The Money Market Protocol. https://compound.finance/documents/Compound.Whitepaper.pdf. [Online; accessed 8-March-2022].

[7] Aave V3: Open-Source Liquidity Protocol. https://docs.aave.com/developers/getting-started/v3-overview. [Online; accessed 8-March-2022].

[8] ChainLink: A Decentralized Oracle Network. https://chain.link. [Online; accessed 8-March-2022].