Using Multi-dimensional Bonding Curves To Create Stablecoins

Bonding curves and their applications in DeFi have been a topic of active research lately. Here, we introduce a novel design of algorithmic stablecoins that is based on bonding curves and solves several issues in previous stablecoin designs. We call this new kind of stablecoins **bonded stablecoins**.

Unlike their predecessors, bonded stablecoins maintain the peg and don't risk becoming insolvent even in the event of abrupt depreciation of the reserve currency, don't require users to overcollateralize, and come with a companion token which is an interest-bearing investment instrument.

Like other DeFi applications that we introduced before -- discount stablecoins, ODEX decentralized exchange, and decentralized token registry -- bonded stablecoins are powered by Autonomous Agents (AAs).

One-dimensional bonding curves

First, a short recap of bonding curves.

The general idea behind bonding curves is to allow minting of some new token in exchange for a reserve asset, with minting happening strictly according to a mathematical formula -- a curve -- that connects the total supply of the token issued and the total amount of the reserve deposited to back the issue. The opposite operation -- redemption of the token for the reserve currency -- happens according to the same formula.

The issue and redemption of the token as well as storage of the deposited reserve is managed by an Autonomous Agent (similar to a smart contract on Ethereum).

The simplest and most well-known case is a one-dimensional bonding curve that links one reserve asset with one token to be issued. The curve in this case is a function of one argument, e.g.:

$$r = s^2$$

where

- r is the total reserve deposited
- s is the total supply of the token issued.

The price per token in units of the reserve asset is the derivative of the above formula:

$$p = dr/ds = 2\ s$$

Note that in this example the price of tokens grows as their supply grows. Therefore, earlier investors get a better (lower) price. This also means that those who are earlier to sell get a

better (higher) price while the next seller will sell their tokens cheaper. Therefore, in the absence of other incentives, the supply is inherently unstable as the holders rush to sell as fast as possible and drive the supply to zero.

For a more thorough introduction to bonding curves, including their application to curation markets, read this article by Simon de la Rouviere as well as other articles by Simon on this topic.

Multi-dimensional bonding curves

Now, we generalize the one-dimensional bonding curves above by introducing more than one dimension.

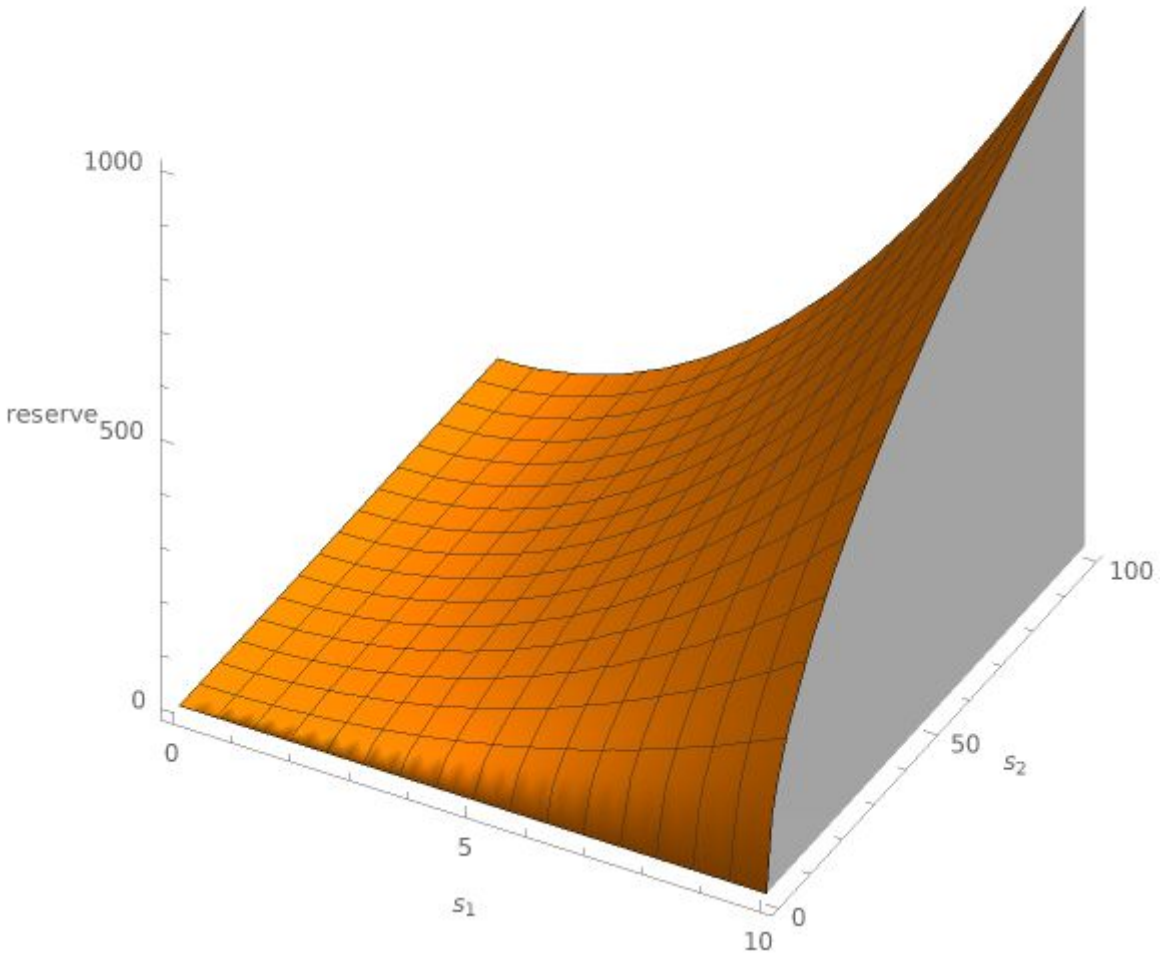
Balancer was first to introduce bonding surfaces by allowing more than one reserve currency to back the issuance of a token. This token is an investment token that represents a portfolio of several reserve currencies in certain proportions. Actually, Uniswap (and Oswap -- its cousin on Obyte) is a special case of such a surface with two reserve assets having 50/50 shares in the portfolio.

We'll go in another direction and construct a multi-dimensional bonding curve that issues *several tokens* against a *single reserve*. The simplest example is a two-dimensional curve (surface) that issues tokens T1 and T2:

$$r = f(s_1, s_2)$$

where

- r is the amount of the reserve asset deposited on the bonding curve AA;
- s_1 is the supply of token T1;
- s_2 is the supply of token T2.



In practical terms, this curve means that one has a choice to issue either token T1 or token T2 (or both at the same time) in exchange for adding some amount of the reserve currency to the bonding curve AA. One needs to specify how many T1 or T2 tokens they want to issue and then send the corresponding reserve delta to the AA. For example, when issuing Δs_1 of token T1, one needs to increase the reserve by

$$\Delta r = f(s_1 + \Delta s_1, s_2) - f(s_1, s_2)$$

The prices of tokens T1 and T2 are partial derivatives of r with respect to s_1 and s_2:

$$p_1 = \partial f / \partial s_1$$

$$p_2 = \partial f / \partial s_2$$

Both prices change in response to changing supplies s_1 and s_2. Now there are more moving parts in this system compared with one-dimensional bonding curves, and this opens new possibilities for financial engineering, in particular for constructing stablecoins.

Pegging the price

Let's say we want one of the prices to be constant, let it be p_2 -- the price of token T2. With one-dimensional bonding curves, by fixing the price of the token, we would have to also fix its supply, which is not interesting. With two-dimensional bonding curves, the price p_2 is a function of two supplies s_1 and s_2, which allows us to keep the price constant at different supplies s_2 by adjusting the supply s_1 of the other token accordingly.

For example, let the bonding curve be

$$r = s_1^2 \, s_2^{½}$$

Then, the prices of tokens are (partial derivatives):

$$p_1 = 2 \, s_1 \, s_2^{½}$$

$$p_2 = ½ \, s_1^2 \, / \, s_2^{½}$$

To keep the price p_2 constant while s_2 increases, one needs to increase s_1, i.e. buy more T1 tokens from the curve. If s_2 decreases, one needs to sell some T1 to the curve to keep the price constant.

So, with two-dimensional bonding curves, it is now possible to keep the price of one of the tokens constant while having its supply flexible. But why would traders buy and sell T1 to return p_2 to the peg? We need to offer them an incentive to do the right thing, and a disincentive to do the wrong thing.

The capacitor

Until now, we assumed that all trades happen according to the curve formula and therefore are completely reversible. Now, let's introduce a fee that will be charged from all trades that push the price p_2 away from the peg. If it was on-peg and you buy or sell T1 or T2 and your transaction changes the supplies s_1 and s_2 in such a way that the price goes off-peg, you pay a fee in addition to what you normally would pay to buy/sell from the curve. If the price was already off-peg and your transaction moved it even further away from the peg, you pay a fee again. The further away from the peg, the larger the fee. This is a disincentive for doing the wrong thing.

All fees charged from such trades are accumulated on a so-called *capacitor* -- a buffer of the reserve currency that is stored on the same AA but is separate from the reserve.

Whenever someone's trade moves the price p_2 back to the peg, they get a reward, which is paid from the capacitor. This is an incentive for doing the right thing.

There is one more thing. Assuming p_2 targets USD price and the reserve currency is GBYTE, the target price of token T2 can fluctuate even when there are no transactions on the curve, just because the reserve asset itself is volatile. Therefore, to make sure that p_2

follows the target, it would be great to always have some juice in the capacitor in order to incentivize the trades that correct the price.

For this reason, we split the capacitor in two parts: slow and fast. All collected fees are distributed between the slow and fast capacitor in some proportions, for example 50/50. But the rewards are paid from the fast capacitor only, while the slow one stays intact. After some timeout, e.g. 3 hours, some share of the slow capacitor, e.g. 10%, will be moved to the fast capacitor, thus refilling it. After another timeout, another 10% of the remaining slow capacity will be moved, and so on. With this procedure, the accumulated capacity will be used over longer periods of time and provide incentive for corrective movements.

The formula that we chose for the fee charged when a trade moves the price away from the target:

$$fee = \mu\, r\, (d_{new} - d_{old})\, (d_{new} + d_{old})$$

where

- μ is a multiplier that specifies how strongly the fee reacts to price deviation;
- r is the average reserve during the trade;
- d_{old} and d_{new} are the old and new distances from the target price.

Distances are defined as

$$d = abs(p_{2,\,target} - p_2) \,/\, p_{2,\,target}$$

where $p_{2,\,target}$ is the target price.

With this formula, the fee is proportional to the distance increment $d_{new} - d_{old}$ and it also becomes steeper as the distance increases (thanks to $d_{new} + d_{old}$).

The reward that is paid for moving the price toward the target is defined by the formula:

$$reward = fast\_capacity\, (d_{old} - d_{new}) \,/\, d_{old}$$

It is proportional the percentage of price correction and would burn the entire fast capacity if the price were returned exactly to the target.

These formulas are somewhat arbitrary; other formulas with similar properties would probably have roughly the same effect. A more advanced and better optimized capacitor can probably be built using a PID controller.

A semi-decentralized protocol Futureswap was an inspiration for the capacitor design as it incentivizes the traders to perform the “right” trades using fees that grow as the disbalance between short and long positions increases (dynamic funding rate). Derivatives exchanges such as BitMEX also use a fee/reward mechanism (funding rate) to keep the prices of their perpetual futures near the price of the underlying asset.

Note that the fee and reward effectively change the price at which tokens are bought or sold. If the fee applies, the trader gets a worse price than the curve would suggest. If the reward applies, they get a better price. The fee and reward are not necessarily equal, therefore buy

and sell prices become different when the price p_2 diverges from the target. A significant difference could lead to the emergence of secondary markets where tokens are traded bypassing the curve (e.g. on ODEX decentralized exchange or on centralized exchanges) while the price stays away from the target for a long time. This situation should be avoided by choosing a curve that also provides incentives for fast elimination of any deviation from the target price, see the next chapter.

Instability of price deviation

The curve itself also provides incentives for fast closing of any deviation from the target price, and an incorrectly constructed curve could in fact provide disincentives.

Recall that the prices of tokens on our curve are:

$$p_1 = 2\, s_1\, s_2^{\frac{1}{2}}$$

$$p_2 = \tfrac{1}{2}\, s_1^2 / s_2^{\frac{1}{2}}$$

Assume for example that there was momentarily a spike in demand for the stable token T2, its supply s_2 increased, which resulted in the price p_2 going down, below the target, according to the second formula above. According to the same formula, the price can be returned to the peg in one of two ways: either by minting more T1 tokens and increasing their supply s_1, or by redeeming some T2 tokens, thus decreasing their supply s_2.

The former way *is* incentive-compatible because by increasing s_1, one also increases the price of T1 tokens p_1 according to the first formula. So, traders rush to buy more T1 before its price has risen, and their buying makes it rise while also correcting the price of the other token p_2.

The latter way *is not* incentive-compatible because in order to correct the price p_2, T2 holders would need to sell T2 while it is cheap. Their selling would drive the price up (according to the curve), which results in them having had a bad deal.

Thus, any deviations from the target price are likely to be corrected by transactions with T1 in the first place. As s_2 grows, there is a strong incentive for s_1 to grow in sync. When s_2 falls, s_1 gets redeemed too.

For a fast return to the target price, it is sufficient that price-correcting transactions with at least one token be incentive-compatible. On our curve, it is T1.

It is possible to design curves that are not incentive compatible in either token, for example, for the curve

$$r = s_1^{\frac{1}{2}}\, s_2^{\frac{1}{2}}$$

the prices of tokens are:

$$p_1 = \tfrac{1}{2}\, (s_2 / s_1)^{\frac{1}{2}}$$

$$p_2 = \tfrac{1}{2}\, (s_1 / s_2)^{\frac{1}{2}}$$

Again, if s_2 increases, it pulls p_2 down, below the peg. To correct the price, one needs either to buy T1 or sell T2. By buying T1, one also decreases p_1, which is not a good idea as it will depreciate in value. Additionally, by selling T2, one increases p_2 -- again a bad trade as it appreciates.

This conclusion also applies to a subset of power-law bonding curves

$$r = s_1^m \, s_2^n$$

with $m < 1$ and $n < 1$ as the prices of both tokens (which are equal to partial derivatives with respect to the corresponding supply) have the supply in the denominator.

So, these curves are not incentive-compatible. The price stabilization incentive provided by the capacitor still works but it works against the incentives that follow from the curve itself. Using such a curve for stablecoins is probably not a good idea as a well-engineered system should avoid unnecessary tension between its parts.

Therefore, curves with either $m > 1$ or $n > 1$ are recommended for stablecoins.

Arbitrage

It is expected that the existence of the capacitor alone will make traders believe that the price will follow the peg. Therefore, any deviation from the peg presents an arbitrage opportunity, which traders will rush to take advantage of, thus returning the price to the peg. If this reasoning works, the exact realization of the capacitor is not important, though we have yet to see this in practice.

The target price $p_{2,\,\text{target}}$ is supposed to be periodically posted by an oracle. For example, for a USD-pegged stablecoin and GBYTE reserve, we'll need an oracle that posts the GBYTE/USD price. Such an oracle already exists and posts the price every 10 minutes. Between the oracle postings, the target price can change, and the arbitrageurs are expected to move the price p_2 to what they expect the next oracle posting will be, i.e. to the target price. This will happen even before the new target price becomes known to the AA and the capacitor mechanism gets activated. Even if the oracle temporarily fails to post the next price or mistakenly posts a wrong price one or more times, arbitrageurs are still expected to keep the price near the peg as long as they believe that the oracle will recover from failure.

Interest

Now we know how to make the price of token T2 stable. It follows from the above that it is the actions of profit-seeking holders of T1 what makes T2 price stable. Note however that T1 holders profit not only from trades that return T2 price to the peg, they also profit from the growth of the ecosystem, i.e. from increasing supply of the stable token s_2. Indeed, from the equation for p_2

$$p_2 = ½ \, s_1^2 \, / \, s_2^{½}$$

we can calculate s_1 as

$$s_1 = (2\ p_2\ s_2^{1/2})^{1/2}$$

and insert it into the equation for p_1:

$$p_1 = 2\ s_1\ s_2^{½} = 2^{3/2}\ p_2^{1/2}\ s_2^{3/4}$$

Since p_2 is supposed to be constant, p_1 rises with the growth of s_2. Therefore, T1 holders are interested in the growth of the stablecoin supply.

However, stablecoin uses in crypto are still limited while there is already a lot of competition in the market, and a new stablecoin has to offer something else to be adopted.

The solution is in adding interest payments to T2 holders. Instead of targeting the price of a fiat currency such as USD, p_2 would target USD price plus some interest that accumulates over time. For example, if interest is 10% per annum, the target price would be 1 USD now, 1.10 USD in one year, 1.21 USD in two years (the interest is compounding), and so on. The target slowly grows over time and this coin can be called a **"stable+" coin**.

This interest-earning opportunity would make it profitable to hold T2 and would attract interest-seeking investors into T2. This is exactly what T1 holders would want in order to grow the price of their token even faster.

However, since the T2 price now grows (relative to USD), it stops being a fiat-pegged coin and is no longer as attractive as a unit of account. Something that grows in value is also not a good *currency* -- according to Gresham's law, it will be among the last to be spent.

To solve this issue, we'll have another AA that will accept deposits in T2 tokens and pay the corresponding amounts of another token which is supposed to be really pegged to USD. Some time later, when the deposited T2 tokens have grown in value a bit, the deposit owner will be able to claim some additional pegged tokens from the deposit AA, that is the accrued interest. The deposit owner will be able to close the deposit and get the deposited T2 tokens back at any time by returning the same amount of the pegged tokens they received against the deposit.

Thus, the interest payments can be stripped from interest-bearing T2 tokens.

The deposit holder also has the option to assign somebody else as the recipient of interest payments, thus donating these interest payments to them. So, one can use the mechanism to support a good cause by donating interest payments to a charity.

For USD-pegged stablecoins, we'll call interest bearing T2 tokens IUSD (interest USD) and stable value token OUSD (open USD). OUSD will be equal to 1 USD and can be used for regular payments, as a quote currency in exchanges, and as a stable unit of value in other AAs and smart contracts where value stability is important.

What makes OUSD keep the peg at 1 USD? If the price of OUSD goes above 1 USD then it is profitable for IUSD holders (recall that the capacitor holds the price of IUSD at 1 USD plus

interest) to exchange IUSD for the overpriced OUSD by opening new deposits and then sell OUSD on the market. This puts sell pressure on OUSD and drives its price down, back to 1 USD.

If the price of OUSD goes below 1 USD, the deposit holders can buy the underpriced OUSD from the market in order to close their deposits and get the fairly priced IUSD in exchange, thus putting buy pressure on OUSD and driving its price up. However, this motivation might not be sufficient as deposit holders might still cling to their deposits that earn them interest and be reluctant to close them.

To ensure price stability, we allow third parties to close other people's deposits by just sending OUSD to the deposit AA and indicating the deposit they want to close. The closing of a deposit, although it doesn't constitute a direct loss to the owner, would stop the flow of interest payments to the deposit owner, and the owners obviously don't want their deposits to be closed. To protect their deposits from closing, the owners have an option to add a *protection* in the reserve currency (GBYTE by default) to the deposit, and only the least protected deposit is allowed to be closed by third parties. The protection is automatically returned to the owner when the deposit is closed. The owner can also withdraw the protection, or part of it, at any time but that would weaken the deposit protection ratio (defined as protection amount divided by deposit amount), and could put their deposit at risk of becoming the first in the line to be closed by third parties the next time OUSD price temporarily dips below the peg.

It is expected that the need to keep a protection next to each deposit would bind even more GBYTE (or other reserve currency) in the AAs.

The bonding curve AA that issues T2 tokens can be instructed to immediately send them to the deposit AA, so the user can immediately convert from GBYTE to OUSD. It is not difficult to also automate conversions from fiat currencies and other cryptocurrencies to GBYTE and immediately send GBYTE to the bonding curve AA, so users can easily start using Obyte-based applications that would benefit from a stablecoin, such as PolloPollo.

If the T2 token targets something different from a fiat currency, such as BTC, gold, another commodity, share price, a stock index such as S&P500, etc., then adding interest payments is not necessary to attract investors into T2 as T2 already has an investment or speculative (opportunity of growth) component.

Multiplication of value

When tokens are issued on a bonding curve, the total value (market cap) of the issued tokens is not necessarily the same as the value of the reserve locked to issue them.

On our curve, the market caps c_1 and c_2 of tokens T1 and T2 respectively are:

$$c_1 = p_1 s_1 = 2 s_1^2 s_2^{½} = 2 r$$

$$c_2 = p_2 s_2 = ½ s_1^2 s_2^{½} = ½ r$$

and the total market cap of both tokens is 2.5 r, that is 2.5 times greater than the locked reserve. That's how bonding curves allow to multiply the amount of value in circulation and benefit the holders of both tokens. In other words, issuing tokens on such a bonding curve is a positive-sum game.

This is true for the specific bonding curve we are considering but it is not always the case. For a more general power-law bonding curve

$$r = s_1^m s_2^n$$

prices are partial derivatives with respect to the corresponding supplies:

$$p_1 = m s_1^{m-1} s_2^n$$

$$p_2 = n s_1^m s_2^{n-1}$$

and market caps of the tokens:

$$c_1 = p_1 s_1 = m s_1^m s_2^n = m r$$

$$c_2 = p_2 s_2 = n s_1^m s_2^n = n r$$

Hence the total market cap is

$$c = (m + n) r$$

This means that if $m + n = 1$, e.g. $m = ½$, $n = ½$, then the game gets zero-sum. In this case, the wealth gets re-distributed between the holders of the two tokens as the target price changes. For example, if T2 tokens are USD-pegged, their holders are taking short positions in the reserve currency (GBYTE) while holders of T1 tokens are taking leveraged long positions in GBYTE. As noted above, powers less than 1 are not incentive-compatible and it is only the capacitor that pushes the price back to the peg.

Getting back to our curve with $m = 2$, $n = ½$, note that the market cap of the stable token (or interest token if interest rate is non-zero) is half of the reserve, this can be interpreted as the stable token being 200% collateralized. However, that doesn't mean that buyers of the stable token need to overpay. They pay exactly as much as the acquired tokens are worth and the other 100% are provided by T1 buyers when they return the price to the peg. T1 holders also make much riskier bets on the growth of the ecosystem and are rewarded with their tokens being worth two times the reserve.

Leverage

Above, we were using bonding curves to create tokens whose price is either stable relative to another currency, commodity, index, etc., or targets a currency plus some interest that accrues over time.

However, the possibilities of bonding curves do not end here, we can also target any synthetic "price" feed which is derived in any way from the existing price feeds.

One simple example is using a price squared, a price^3, or a price in any other power instead of the regular price. A T2 token that targets such a synthetic price would offer a kind of leveraged exposure to the underlying asset. Indeed, if we use a squared price and the underlying price rises by 1% then the squared price rises by 2.01%, which is approximately what a 2x leverage would yield.

However, this kind of leverage (let's call it **power-law leverage**) differs from the classic leverage:

- To take a classic leveraged position, one needs to borrow. For example, in 2x leverage, the size of a position is 200% of own money where half of the total position is own money and the other half is borrowed. In power-law leverage, one just buys tokens that are more volatile than the underlying asset, there is no debt.
- In classic leverage, the relative change in one's net wealth per relative change in price is not constant. It changes as the price goes further away from the initial price. For example, with 2x leverage, every 1% change in price near the initial price yields 2% change in net wealth. But when the price rises 3x, net wealth rises 5 times (2*3 minus 1 borrowed part) and further increase of price by 1% increases the position value by 6% which is only 6%/5 = 1.2% of the prior net wealth. So, the leverage relative to net wealth decreases as the price grows. Conversely, the relative leverage increases as the price goes down. In power-law leverage, on the opposite, the relative change in one's net wealth per relative change in price *is* constant, at any price. Power-law leverage can be emulated using classic leverage if one borrows more against the increasing net wealth as the price goes up, and repays part of the borrowed capital as the price goes down to keep the ratio between the borrowed and own capital constant.
- In classic leverage, one can be liquidated and lose the entire capital if the price goes down too far. In power-law leverage, it is impossible, as the trader survives depreciations of any magnitude. This means that if a trader is correct about long-term prospects of an asset and takes a leveraged position, then under classic leverage they can still be burnt by short-term price movements against them. With power-law leverage, they can just buy and hold without fear of being liquidated.

So, one can define a bonding curve where the price of T2 token is stable relative to a squared (or another power of) price of a target asset. This kind of stability is relative, it is not a stability of purchasing power as one might expect.

Adding interest payments to leveraged tokens is not necessary as they are already an investment instrument.

In the web application that we built to help issue bonded stablecoins, the default reserve currency is GBYTE and the price feed provided by the price oracle is GBYTE/USD, not USD/GBYTE. That's why some relations are reversed and the leverage is defined as leverage in reserve currency (GBYTE) vs USD. Hence,

- leverage = 0 means tracking USD price;
- leverage = 1 means tracking the reserve (GBYTE) price;

- leverage = 2 means 2x long position in the reserve currency (GBYTE) relative to the oracle price (USD);
- leverage = -1 means taking short position in the reserve asset (GBYTE) relative to the oracle price (USD);
- leverage = -2 means taking 2x short position in the reserve asset (GBYTE) relative to the oracle price (USD).

Note that taking any position, including a short one, requiries locking up some GBYTE as reserve, which reduces the free float of GBYTE and drives the price up, so taking a short position this way might be seen as self-contradictory.

Governance

There are several parameters that might need to be adjusted to ensure stability and usability of a stablecoin and encourage its widest possible adoption.

Among these parameters:

- **Interest rate paid to T2 token holders.** It should be attractive enough compared with other investments, but still sustainable.
- **The oracle whose data is used to target the price of the stablecoin.** Obviously, its data should be true and accurate and the oracle's operation should be reliable (i.e. it should not suddenly disappear).
- **The multiplier used for fees charged when the price goes away from the target.** It should be large enough to discourage large deviations from the peg but not too steep, so as not to make buyers pay large fees for small transactions that only slightly move the price away from the peg.
- **How the fees are split between the fast and slow capacitors.** We need a strong enough fast capacitor to ensure immediate response to any price deviations, but at the same time we don't want to spend all accumulated fees immediately and want to save something for future rewards for price-correcting trades.
- **Threshold deviation from the target price that activates the flow of funds from the slow to the fast capacitor.** We want to keep the price tightly near the peg, but also want to save the slow capacity for a rainy day.
- **How fast the funds flow from the slow to the fast capacitor.** We need to refill the fast capacitor in time to ensure sufficient rewards for price-correcting trades but we also want to save for the future.

The holders of the T1 token have skin in the game as the price of their token depends on the level of adoption of the stablecoin, and they have the power to make decisions about these parameters. For a USD-pegged stablecoin, we'll call the T1 token GRD (**gr**owth **d**ollar).

T1 holders vote with their tokens and the vote weight is proportional to the amount of T1 tokens they deposit on a governance AA that is designed specifically for this purpose. There can be several proposed values for each parameter and those who vote for the winning value cannot withdraw their T1 tokens for 30 days after the voting ends to make sure that they cannot sell and are fully exposed to the outcome of their decision.

To determine the winner, it would be impractical to require the majority of all T1 holders to vote. Instead, we use a voting method that we call **challenge voting**.

When any proposed decision becomes a leader (i.e. it has more votes than any other competing decision), it enters a challenging period. The period is 3 days for the less important parameters and 30 days for the more important ones such as the choice of the oracle (both periods are configurable and can be set to different values when creating the stablecoin AA). During the challenging period, the supporters of the competing decisions have the time to mobilize and try to overtake the leader by votes. If this happens, another decision becomes the leader and the voting enters another challenging period. Supporters of the leading decision are able to add support to their decision during the challenging period in order to prevent being overtaken. If the challenging period expires without the leader being challenged, it becomes the accepted decision and the newly decided value is activated in the bonding curve AA.

Challenge voting makes it possible that only active, dedicated stakeholders need to participate in decision making while others don't intervene as long as the decisions are not important enough or the passive stakeholders trust the more active voters to know what they are doing. Thus, decisions are made quickly, without undue delays, while the passive stakeholders always have an opportunity to join the process if they feel the decision is important.

Management team

While T1 holders are interested in the success of their stablecoin, most of them are just investors and cannot dedicate much time to promote the use of the stablecoin, such as working with exchanges and fiat gateways to get it listed, working with merchants to get it accepted, and so on. Most of them also do not have the skills required to do this work.

They might want to hire a professional management team for this job. The governance AA allows them to do so.

A prospective team would publish their proposal stating that they want to be hired for a specific term (e.g. 1 year), would do this and that, expect to achieve this and that, and the amount in T1 tokens they want to be paid for their work. Maybe another competing team would publish their own competing proposal. The stakeholders would then vote for the proposal they like.

This is a very important decision and the approval of 50% (or some other threshold specified during creation of the AA) of the stakeholders is required to accept a proposal. Once a proposal is accepted, new T1 tokens will be issued and sent to the team that won the vote and the bonding curve formula will be modified to account for the increased supply of T1 tokens without affecting the price of the stablecoin p_2. Thus, the T1 holders will be diluted in exchange for the promise of the winning management team to develop the ecosystem. For example, if the winning team requested a reward that is 1% of the existing T1 supply, the existing T1 holders will be diluted by 1% and the price of their tokens (according to the

bonding curve) will fall by 1% on the day when new T1 tokens were printed and sent to the winning team.

Thus, T1 holders can act like shareholders and elect the executives who manage day-to-day operations, report about their performance, and want to be hired for the next term.

The stakeholders might decide that they don't need a management team, but if they do, that'll be the power that emerges from the community.

Other bonding curves

We mostly focused on one particular two-dimensional bonding curve

$$r = s_1^2 \, s_2^{½}$$

but infinitely more bonding curves are possible. Not all of them would be good for stablecoins, however.

We've already mentioned some properties of the power-law bonding curves

$$r = s_1^m \, s_2^n$$

Let's see how different values of m and n would determine the properties of the stablecoin if, like before, the T2 token price targets the price of some asset such as USD.

Recall that the prices of tokens on such a curve are:

$$p_1 = m \, s_1^{m-1} \, s_2^n$$

$$p_2 = n \, s_1^m \, s_2^{n-1}$$

As we've seen above, we need either $m > 1$ or $n > 1$ for the curve to be incentive-compatible.

If we take $n > 1$, then both factors in the expression for p_2 have positive powers. Therefore, in order to keep p_2 constant as s_2 grows, T1 holders will have to sell their tokens. There are two issues with this scenario:

- T1 holders would be reluctant to sell a token that is supposed to grow and this would create tension on the peg;
- as T1 supply shrinks, the user base of T1 holders is likely to shrink too, so the token would only be held by a small and closed group of people.

To avoid that, curves with n < 1 are preferred as the power of s_2 in the expression for p_2 is then negative. Hence, s_1 needs to grow to accommodate the growth of s_2 and new members have the opportunity to join the community of T1 holders.

With $n < 1$, in order for the curve to be incentive compatible, the other power, m, has to be greater than 1. Our chosen curve with $m = 2$, $n = ½$ belongs to this family.

If we use the above two equations to express p_1 through s_2, we get:

$$p_1 = m\ (p_2 / n)^{(m - 1) / m}\ s_2^{\ 1 + (n - 1) / m}$$

We can see how the choice of m and n affects the price trajectory of p_1 with the growth of s_2. For example, if we take a curve with $m = 3$, $n = ½$, we get a slightly steeper growth of p_1 (compared with our standard curve), but not much steeper.

An interesting case is a curve with $m = 1$, $n = 1$:

$$r = s_1\ s_2$$

Then the prices are

$$p_1 = s_2$$

$$p_2 = s_1$$

The curve has a good property that buys/sells of the stable token T2 do not affect its price at all. However, the community of T1 holders does not grow and if p_2 were to fall, then the community would need to shrink. This is because for a USD-pegged stablecoin and GBYTE reserve, p_2 is USD/GBYTE price, the reverse of GBYTE/USD quoted by exchanges, and USD is likely to fall vs GBYTE as more GBYTE gets locked for the reserve.

Other non-power-law bonding curves are also possible, such as more general polynomial curves, exponential curves, and actually anything math can do. They present an interesting area of research for stablecoin and other applications, and with this article, we have probably scooped only a small part of what is possible.

Bonding curves on DAG vs blockchain

One can think about porting multi-dimensional bonding curves and bonded stablecoins to a blockchain-based platform such as Ethereum. However, they would be vulnerable to various manipulations:

- **Miner manipulation**. Every trade on a bonding curve changes the prices, and being later might mean getting a worse price. Miners (or rather mining pools) have control over the order of transactions in the block and can include their own transaction before a user's transaction, so the user will transact at a worse price than they expected, could pay a high fee for pushing the price of T2 token off the peg, and the miner will then be able to send the opposite transaction that corrects the price and earns them a reward. This is a typical example of front-running, which is a recognized issue in centralized finance and many brokers have been caught committing this kind of abuse. Preventing such abuse is one of the reasons why centralized finance needs regulation. Miners are centralized entities like brokers and their outsized control over the composition of blocks makes front-running a real concern for blockchains as well. On Ethereum, the issue is exacerbated by extreme centralization of mining -- just two mining pools mine more than half of blocks now.

On a DAG-based ledger like Obyte, there are no miners and nobody could realistically insert their transaction before yours.

- **Non-miner manipulation**. Even assuming miners don't play such games (e.g. they are regulated by SEC, cannot create anonymous accounts, etc.), non-miners can manipulate the order of transaction as well -- just by paying a higher fee to a miner. The higher fee is like a bribe and it ensures that the transaction paying a higher fee is included earlier in the blockchain than the other transaction, thus the other transaction gets a worse price again even if it was received earlier. On a DAG, again, there is no way to change the order of transactions that are already added to the DAG.

Thus, bonded stablecoins on a DAG would be much safer than bonded stablecoins on a blockchain. Now, let's compare bonded stablecoins against other crypto collateralized stablecoins focusing on their properties that are independent of the platform they are running on.

Comparison with other stablecoins

Compared with other crypto collateralized stablecoins such as DAI, the main advantage of bonded stablecoins is that it is easy to issue them without overcollateralizing. Users need to pay exactly as much as the purchased stablecoins are worth. The rest of the reserve is contributed by investors who bet on the success of the stablecoin.

Also, the pegging mechanism of bonded stablecoins continues working even under wild volatility conditions with abrupt price movements. DAI and similar stablecoins on the other hand, cannot survive fast and significant depreciation of the collateral asset and they would become insolvent. The threat is mitigated with overcollateralization but no overcollateralization is enough for all market conditions.

Bonded stablecoins are also not vulnerable to low liquidity of the reserve asset. Illiquid markets can be easily manipulated to cause temporary depreciation of the reserve asset and hence undercollateralization of the loans in DAI-like stablecoins, which would make them auctioned off and cause losses for the loan holders. In bonded stablecoins, there are no loans, no minimum collateralization requirements, and such manipulations do not make sense.

Discount stablecoins were our previous generation stablecoins that are based on overcollateralized loans, much like DAI, and also appreciate in value similar to zero-coupon bonds (we later discovered that Dan Robinson of Paradigm independently came up with a very similar design which he called yTokens, however nothing has been launched yet). They are an investment product whose value is not exactly stable and like bonds, they expire. Bonded stablecoins, on the other hand, do not expire and allow to have both an interest-bearing token such as IUSD and a stable USD-pegged token such as OUSD.

Bonded stablecoins are already live

Head to ostable.org to issue or redeem bonded stablecoins, open or close interest bearing deposits, participate in governance, or even launch new stablecoins on another bonding curve or pegged to a different target asset.

Bonded stablecoins replace discount stablecoins on this website, and discount stablecoins have been moved to discount.ostable.org.

USD-pegged stablecoins are already there. You can buy interest-bearing IUSD tokens (stable+ coins), USD-pegged OUSD tokens, and the growth token GRD that grows as the supply of the IUSD token grows.

You can use this website to launch new bonded stablecoins that target other fiat currencies, gold, other commodities, shares, stock indexes such as S&P500, and actually anything that has a numeric value, has public significance, and can be reported by an oracle (how about the number of twitter followers of Obyte for example?). Stability of these stablecoins is *relative to their target* and when the target is volatile (in terms of a “really stable” value such as USD), the stablecoins will be equally volatile.

By creating stablecoins pegged to various real-world and synthetic assets, one can enable trading of these assets by a broader audience of investors worldwide without needing to access the actual markets and using intermediaries such as brokers.

Currently, there are oracles that post the prices of fiat currencies, cryptocurrencies, and precious metals, and the corresponding stablecoins can be launched immediately. To track other targets, anyone can set up a new oracle that posts the corresponding prices, indices, etc. There is an example of an oracle source code on GitHub one can use to start a new oracle.

Bonded stablecoins are served by a family of Autonomous Agents, and their source code is published on our GitHub.

Multi-dimensional bonding curves is a new area of research and experimentation and it is quite likely that many other DeFi products can be built with them. If you have new ideas, you are welcome to discuss them on the Obyte discord and build your revolutionary DeFi app using Autonomous Agents.