Explicit Formulas for Algebraic Sigmoid Bonding Curves

Christopher H. Gorman

November 29, 2022*

1 Introduction

There have been a number of recent articles discussing token bonding curves [2, 4, 8, 9, 10]. The initial type of bonding curve came from the Bancor protocol [6] (originally called "smart tokens"), which focuses on keeping a constant faction of the deposits in reserve. Other bonding curves have been suggested to compensate for the perceived drawbacks of the Bancor bonding curve [3, 7]. In particular, [1, 11] suggest to using *algebraic* bonding curves; we recall that algebraic functions do not use logarithms or exponentials.

Performing complicated mathematical operations in Solidity is possible but may be challenging [12]. No explicit minting and burning formulas for algebraic sigmoid bonding curves are found in [1, 11]. This is unfortunate because bonding curves are useless without formulas which can be readily translated into Solidity code.

In this article we review the purpose of bonding curves. Additionally, we develop explicit formulas for minting and burning tokens based on an algebraic sigmoid bonding curve. In the process, we present a new way of viewing bonding curves which enables the construction of a bonding curve which smoothly varies between two prices.

Note: some of the notation used here is different from other articles on bonding curves. The reason for this is discussed below.

2 Initial Bonding Curve Discussion

2.1 Intuition and Purpose

One of the well-known problems with cryptocurrencies is the large price fluctuations which may occur. These oscillations increase the risk of participating and in so doing decrease adoption. One attempt to decrease this volatility is to have tokens with prices controlled by bonding curves.

The idea behind bonding curves is that the price of minting additional tokens should only depend on how many tokens already exist. Generally speaking, the price of minting more tokens will increase with the total supply. The usual idea behind increasing the cost

[*] Initial version: November 29, 2022

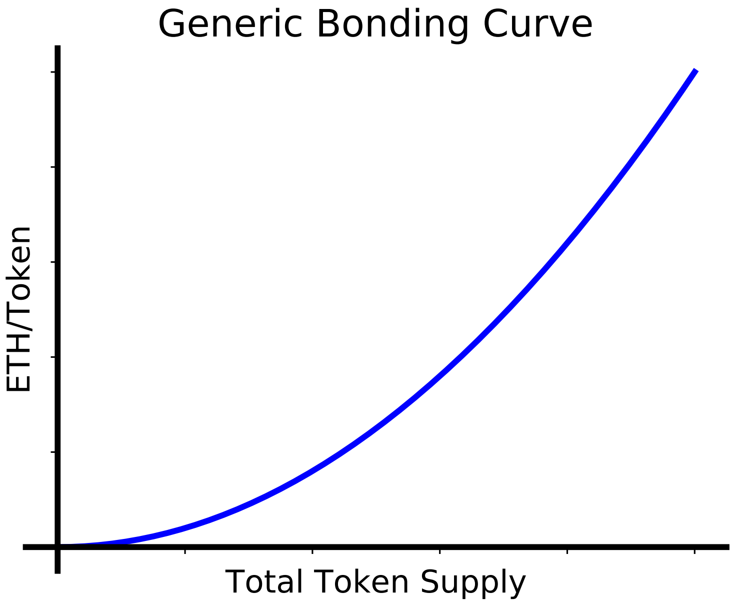


Figure 1: Example of a generic bonding curve. As the total token supply increases, the price (ETH/Token) increases as well. The specifics of the bonding curve specifics the exact change.

to mint additional tokens is to incentivize early adopters. Eventually, the price will increase until it reaches the market price. Upon reaching the market price, no additional tokens will be minted.

Throughout this article we are assuming continuous tokens, so in theory we can mint and burn tokens at arbitrary sizes. This allows us to use calculus. Additionally, in this article we will assume that these operations are performed in Ethereum, so the underlying currency is ETH. All of this may be readily translated to the more general setting. Because of this, we are interested in the token price (ETH per token).

2.2 Setup

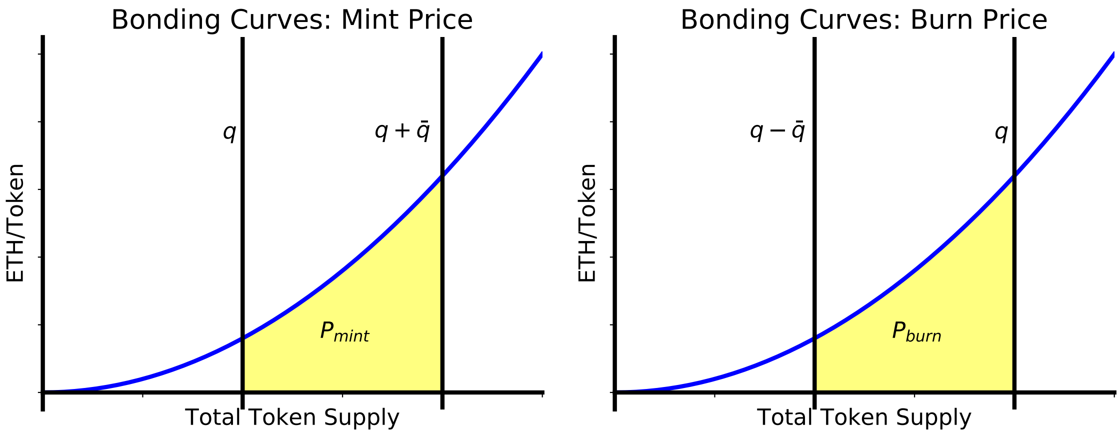
The mint and burn prices are fully determined by the token supply changes. Formally, we define the price function f_{price} where $f_{\text{price}}(q)$ is the price to purchase tokens at total token supply q. For simplicity we assume that f_{price} is smooth (continuous). An example of a generic bonding curve can be found in Fig. 1.

Suppose Alice wishes to purchase $\bar{q}$ tokens and the total token supply is q. Set

$$P_{\text{mint}} = \int_{q}^{q+\bar{q}} f_{\text{price}}(s)\, ds. \tag{2.1}$$

Then Alice must spend P_{mint} ETH to purchase $\bar{q}$ tokens. Similarly, suppose that she wishes to burn $\bar{q}$ tokens. Set

$$P_{\text{burn}} = \int_{q-\bar{q}}^{q} f_{\text{price}}(s)\, ds. \tag{2.2}$$



(a) Mint price for $\bar{q}$ tokens.

(b) Burn price for $\bar{q}$ tokens.

Figure 2: Here are the prices for minting and burning.

Alice then receives P_{burn} ETH for burning $\bar{q}$ tokens. An example showing the price of minting can be found in Fig. 2a while an example showing the price of burning can be found in Fig. 2b. In both cases, it is clear that P_{mint} and P_{burn} depend *only* on the total token supply q, the tokens $\bar{q}$ to mint or burn, and the bonding curve f_{price}.

While in this example the mint and burn curves are the same, it is possible to have different mint/burn curves. By having the burn price below the mint price, the market will be able to have some controlled volatility. In this way, a change in a token's market price does not necessitate a change in token supply. Large deviations above the minting price or below the burning price would lead to arbitrage opportunities; this should help limit volatility. An example with different mint and burn curves can be found in Fig. 3. Different mint and burn curves imply that minting and burning result in different prices; see Fig. 4. In this way, minting tokens and then immediately burning those same tokens results in an economic loss.

2.3 In Practice

The minting and burning example we briefly discussed works well in theory where individuals know exactly how many tokens they desire to mint or burn. We now discuss how minting and burning may work in practice.

2.3.1 Minting in Practice

In practice, Alice may not necessarily know how many tokens she wishes to purchase; instead, she may want to use P_{mint} ETH to purchase some amount of tokens.

To determine how many tokens P_{mint} ETH mints, let f_{price} be as before and set

$$F_{\text{price}}(t) = \int_0^t f_{\text{price}}(s)\,ds. \tag{2.3}$$

We know

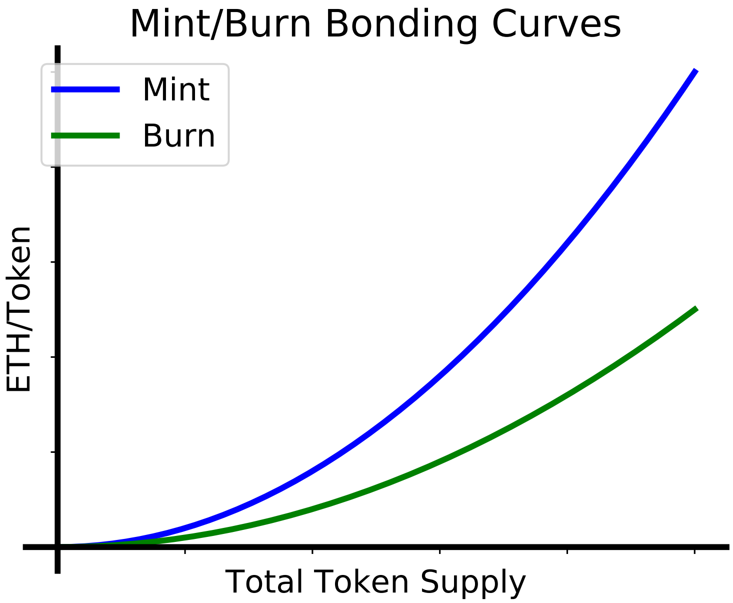


Figure 3: Example of generic bonding curves with different mint and burn prices.

$$\begin{aligned} P_{\text{mint}} &= \int_{q}^{q+\bar{q}} f_{\text{price}}(s)\, ds \\ &= F_{\text{price}}(q+\bar{q}) - F_{\text{price}}(q). \end{aligned} \tag{2.4}$$

This means we must solve

$$F_{\text{price}}(q+\bar{q}) = P_{\text{mint}} + F_{\text{price}}(q). \tag{2.5}$$

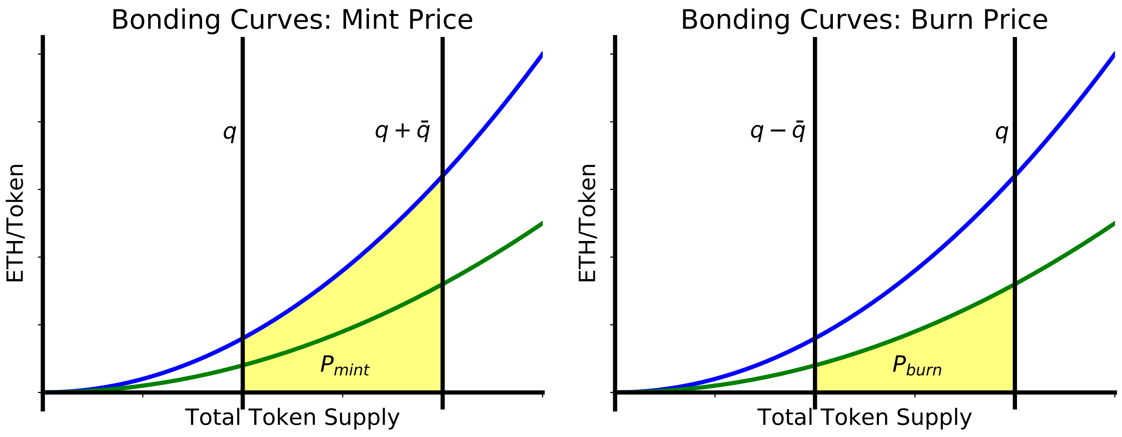
for $\bar{q}$. To do this, we must compute the inverse function of F_{price}. The difficulty of computing this inversion depends on f_{price}.

2.3.2 Burning in Practice

Alice may have $\bar{q}$ tokens she wishes to burn. In this case, she will receive P_{burn} ETH in return, where

$$\begin{aligned} P_{\text{burn}} &= \int_{q-\bar{q}}^{q} f_{\text{price}}(s)\, ds \\ &= F_{\text{price}}(q) - F_{\text{price}}(q-\bar{q}). \end{aligned} \tag{2.6}$$

This computation is easy given F_{price}.



(a) Mint price for $\bar{q}$ tokens.

(b) Burn price for $\bar{q}$ tokens.

Figure 4: Here are the prices for minting and burning for different minting and burning curves.

2.4 Discussion of Practice

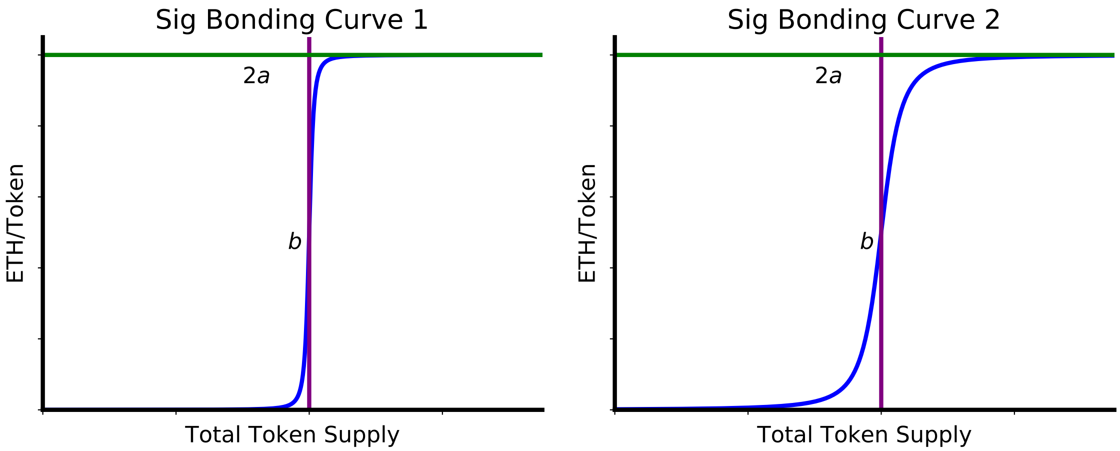
From the above discussion, we see that in practice Alice may wish to spend a certain amount of ETH to mint tokens. While it may happen that she wants to purchase a specific number of tokens, ETH is the standard currency and must be deposited anyways. Focusing on the price as a function of token supply works well *in theory*; this can be challenging in practice because F_{price} must be simple enough in order to accurately invert. The Bancor formula [6] allows for such inversion at the cost of requiring careful calculation involving logarithms. As shown in [13], this is not necessarily a problem, but care is required.

Furthermore, the proper granularity is needed. It is unlikely everyone would be willing to spend one ETH on a new token, so it is useful in practice to have all purchase values be in WEI. As it is certainly possible for someone to spend 1 ETH on a new project, this requires that the necessary granularity for submitted values must be able to handle inputs of size 10^{18} or larger with ease. Given that $2^{256} \simeq 10^{77}$, and $(10^{18})^5 = 10^{90} \gg 2^{256}$, care is required even when using small degree polynomials.

3 Previous Discussion of Sigmoid Bonding Curves

The Bancor formula [6] has some good properties, but as discussed in [3, 7], other bonding curves may be desired in practice. For instance, there has been interest in allowing for sigmoid bonding curves to ensure an asymptotic price; these curves have been discussed in [1, 11]. The preference for an *algebraic* sigmoid bonding curves is to enable greater ease of implementation in Solidity. Note that there is a slight difference in notation between [1] and [11].

The algebraic sigmoid bonding curve has the form



(a) Bonding curve with small c.

(b) Bonding curve with large c.

Figure 5: Here are two examples of sigmoid bonding curves; the form is given in Eq. (3.1). We see the initial price is close to 0 while the price assymptotically approaches $2a$. The transition is marked by b, and c controls the rate of this transition: $f'_{\text{price}}(b) = a/\sqrt{c}$.

$$f_{\text{price}}(x) = a\left[\frac{x-b}{\sqrt{c+(x-b)^2}}+1\right]. \tag{3.1}$$

We set

$$\begin{aligned} F_{\text{price}}(t) &= \int_0^t f_{\text{price}}(s)\,ds \\ &= a\sqrt{c+(t-b)^2}+at-a\sqrt{c+b^2}. \end{aligned} \tag{3.2}$$

Example plots with different c values can be found in Fig. 5. Here, a controls the asymptotic cost (where $2a$ is the asymptotic price), b marks the transition from the initial price region to the asymptotic price region, and c controls how quickly this transition occurs. Larger c values result in a more gradual transition to the asymptotic price. More precisely, we have $f'_{\text{price}}(b) = a/\sqrt{c}$.

These algebraic bonding curves allow for unlimited tokens to be minted and asymptotically approach a pre-specified limit. In this way, the price smoothly varies from close-to-zero until it asymptotically approaches $2a$.

This is great, but for this bonding curve to be used *in practice*, we must be solve

$$F_{\text{price}}(t) = q \tag{3.3}$$

for t given arbitrary $q \geq 0$. More explicitly, we must solve the following algebraic equation for t:

$$a\sqrt{c+(t-b)^2}+at-a\sqrt{c+b^2} = q. \tag{3.4}$$

None of the previously-referenced articles include the solution to this algebraic equation.

For the sake of completeness, we note Eq. (3.4) has the solution

$$t = \frac{1}{2}\left(\frac{q}{a} + \sqrt{c+b^2} + b - \frac{c}{\frac{q}{a} + \sqrt{c+b^2} - b}\right). \tag{3.5}$$

While this is the analytical solution which holds for all $q \geq 0$, the solution does assume that $a, b, c > 0$. Note that some work is required to convert the solution into a form that will work with Solidity. This will involve computing integer square roots; efficient algorithms for computing integer square roots are discussed in [5].

4 New Bonding Curve View

We previously saw that given the standard convention of having price as a function of total token supply, minting tokens requires inverting a function. Burning tokens is easier, of course, but we must first *mint* tokens in order to later *burn* them.

4.1 Change in Focus for Bonding Curves

We now proceed in an analogous manner but from a different angle. In Secs. 2 and 3, the bonding curve was defined by f_{price}, which specified the token price as a function of total token supply. We now will specify the *inverse price* parameterized by the ETH reserve.

Formally, we let $\rho(t)$ denote the tokens per ETH for a total t ETH reserve. Set

$$P(x) = \int_0^x \rho(s)\,ds. \tag{4.1}$$

Thus, if Alice wants to purchase k ETH worth of tokens given r ETH reserves, then Alice will receive T_M minted tokens:

$$\begin{aligned} T_M &= \int_r^{r+k} \rho(s)\,ds \\ &= P(r+k) - P(r). \end{aligned} \tag{4.2}$$

Upon burning T_B tokens, Alice will receive k ETH, where

$$\begin{aligned} T_B &= \int_{r-k}^{r} \rho(s)\,ds \\ &= P(r) - P(r-k). \end{aligned} \tag{4.3}$$

In order to burn tokens, we must be able to invert P.

4.2 Discussion of New View of Bonding Curves

This section suggests another way to view bonding curves: specifying the bonding curve as inverse price (Tokens/ETH) in terms of total ETH reserve. This allows for greater ease for minting tokens at the cost of making burning more complicated. This is the reverse of the situation discussed in Sec. 2.

We think this view of bonding curves allows for an easier way to smoothly vary between two token prices: one token price for early adopters and another token price after market stabilization.

4.3 Desired Features of Bonding Curves

We now discuss some of the features of the new bonding curve we are interested in designing:

- We want to specify two prices: one token price for initial adopters and another token price after sufficiently many tokens are in circulation.
- We want a continuous token model.
- We want the bonding curve to be algebraic; that is, we do not want to use exponentials or logarithms.
- We want a *smooth* change in token price. We also want to be able to control how rapidly the token price changes.
- We want a closed form solution for the minting and burning bonding curves.

Overall, we want an algebraic sigmoid bonding curve which smoothly varies between an initial token price and an asymptotic token price with a closed form equation for minting and burning.

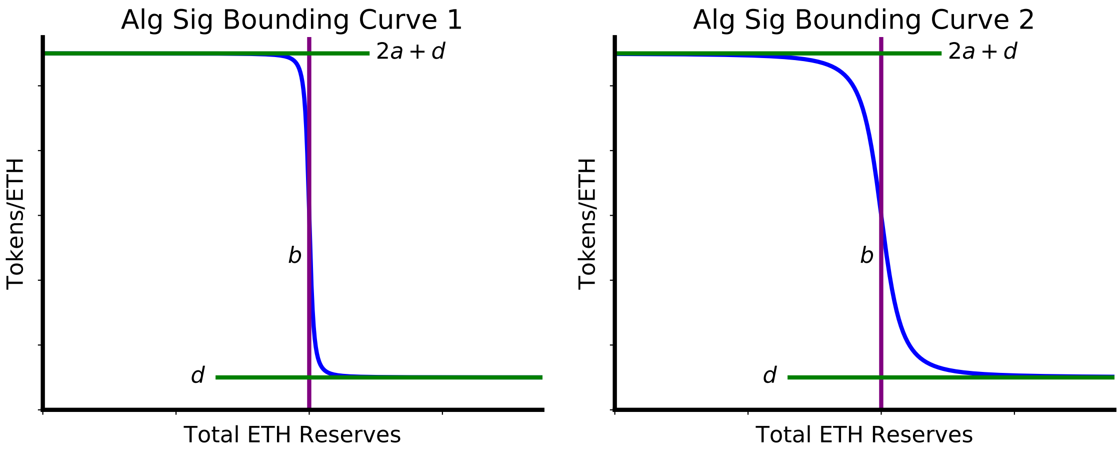
5 Algebraic Sigmoid Bonding Curve

5.1 Bonding Curve Formulas

Here is the sigmoid we are working with for our algebraic sigmoid bonding curve:

$$\rho(x) = a\left[\frac{b-x}{\sqrt{c+(b-x)^2}} + 1\right] + d. \tag{5.1}$$

Here, a, b, c, and d are positive constants. We have that $\rho(x)$ denotes the Tokens/ETH ratio given ETH reserves x; thus, this specifies the inverse price of the Token. Example plots can be found in Fig. 6. We note the following:



(a) Bonding curve with small c.

(b) Bonding curve with large c.

Figure 6: Here are two examples of algebraic sigmoid bonding curves with different c values. Note the y-axis, which specifies Tokens/ETH and the x-axis is in terms of ETH reserves. Bonding curves are usually have the y-axis specify ETH/Token and the x-axis specify total Token supply.

$$\begin{aligned} \rho(0) &\simeq 2a+d \\ \lim_{t\to\infty}\rho(t) &= d \\ \rho(b) &= a+d \\ |\rho'(b)| &= \frac{a}{\sqrt{c}}. \end{aligned} \tag{5.2}$$

We see that

$$\int \rho(s)\,ds = a\left[-\sqrt{c+(b-s)^2}+s\right]+ds. \tag{5.3}$$

We are ignoring the constant factor. If r ETH are in the reserves, the total token supply $P(r)$ is

$$\begin{aligned} P(r) &= \int_0^r \rho(s)\,ds \\ &= a\left[-\sqrt{c+(b-s)^2}+s\right]+ds\Big|_0^r \\ &= a\left[\sqrt{c+b^2}-\sqrt{c+(b-r)^2}+r\right]+dr. \end{aligned} \tag{5.4}$$

As we just mentioned, if r is the total ETH reserves, then $P(r)$ is the total token supply. We also have the inverse: if the total token supply is t, then $P^{-1}(t)$ is the total ETH reserves. Because we can readily evaluate P, we will work toward being able to compute P^{-1}.

5.2 Minting and Burning in Practice

Using the new view of bonding curves, we go through the minting and burning procedures again.

Given r ETH reserves, Alice wishes to purchase k ETH worth of tokens will receive m tokens:

$$m = P(r+k) - P(r). \tag{5.5}$$

This computation is straightforward with P in Eq. (5.4).

If Alice wishes to burn m tokens given r ETH reserve, she will receive k ETH, where the following equality holds:

$$m = P(r) - P(r-k). \tag{5.6}$$

To solve this equation, it is sufficient to always be able to solve $P(r) = m$ for t given arbitrary $m \geq 0$. This is done in Sec. 5.3.

5.3 Inverse Minting Equation and Solution

To more easily compute the burning equation, we solve explicitly for the inverse minting equation. That is, we need to solve $P(t) = m$ for t given arbitrary $m \geq 0$.

Given t ETH in reserves, we have this many tokens:

$$P(t) = a\sqrt{c+b^2} - a\sqrt{c+(t-b)^2} + (a+d)\,t. \tag{5.7}$$

We now wish to invert this for t, so we want to solve

$$P(t) = m \tag{5.8}$$

for t when $m \geq 0$ is arbitrary. This equation can be rearranged to

$$\sqrt{c+(t-b)^2} = \sqrt{c+b^2} - \frac{m}{a} + \left(1 + \frac{d}{a}\right) t. \tag{5.9}$$

Solving for t, we find

$$t = \frac{-c_1 + c_2 m + a\sqrt{d_0 - d_1 m + m^2}}{c_3} \tag{5.10}$$

with

$$\begin{aligned}
c_1 &= a\left[(a+d)\sqrt{c+b^2} + ab\right] \\
c_2 &= a+d \\
c_3 &= d\,(2a+d) \\
d_0 &= \left[(a+d)\sqrt{c+b^2} + ab\right]^2 \\
d_1 &= 2\left[a\sqrt{c+b^2} + (a+d)\,b\right].
\end{aligned} \tag{5.11}$$

This means that we have defined P^{-1} with

$$P^{-1}(m) = \frac{-c_1 + c_2 m + a\sqrt{d_0 - d_1 m + m^2}}{c_3}. \tag{5.12}$$

Given P and P^{-1}, we can use this information to implement this bonding curve in Solidity. This requires computing integer square roots; see [5] for implementation details.

5.4 Formulas for Various Operations

We now list the formulas for various operations. While the variables are generally used consistently, the variables may change in each instance.

Mint Specific Number of Tokens Given ETH

Here was want to determine the number of tokens minted given a specific value of ETH. In particular, if there are r ETH reserves and k additional ETH is used to mint additional tokens, then t additional tokens are minted:

$$t = P(r + k) - P(r). \tag{5.13}$$

This was previously discussed in Eq. (4.2) in Sec. 4.1.

ETH Required to Mint Specific Number of Tokens

Here we want to determine the ETH required to mint a certain number of tokens. While this may not be the "usual case", it is still important.

In particular, if there are r ETH reserves and we want t tokens, then it will cost k ETH so that

$$t = P(r + k) - P(r). \tag{5.14}$$

This is just Eq. (5.13). Solving for k, we find

$$k = P^{-1}\left(P(r) + t\right) - r. \tag{5.15}$$

If we set $T = P(r)$ as the total token supply, then this reduces to

$$k = P^{-1}\left(T + t\right) - r. \tag{5.16}$$

ETH for Burning Specific Number of Tokens

We now suppose that there are r ETH reserves and we want to burn t tokens and want to determine how much ETH we should get in return. In particular, we know

$$t = P(r) - P(r - k). \tag{5.17}$$

This was previously discussed in Eq. (4.3) in Sec. 4.1.

Rearranging and solving for the k ETH returned, we find

$$k = r - P^{-1}\left(P(r) - t\right). \tag{5.18}$$

If we let $T = P(r)$ as the total token supply, then this reduces to

$$k = r - P^{-1}\left(T - t\right). \tag{5.19}$$

Tokens Required to Burn for Specific ETH

We now suppose want to know how many tokens must be burned to receive a specific value of ETH in return. In particular, let r be the ETH reserve and a user wants k ETH. This will require burning t tokens, with

$$t = P(r) - P(r - k). \tag{5.20}$$

5.5 Link to Test Parameters

Here is a link to a Python notebook in order to look at what happens with different parameters. This can be used when determining the appropriate values for a specific bonding curve.

6 Conclusion

In this article we talked about some of the difficulties which arise when working with bonding curves. We also gave an explicit solution enabling one to easily compute with sigmoid bonding curves.

Previously, bonding curves always had token price (ETH per token) as a function of total token supply. Here, we presented a new way to think about bonding curves which focuses on tokens per ETH as a function of ETH reserve. In our situation, we found that this makes it easier to smoothly change between two token prices. We also gave explicit formulae to allow for minting and burning operations.

References

[1] Slava Balasanov. *Bonding Curves In Depth: Intuition & Parametrization*. Dec. 13, 2018. URL: https://blog.relevant.community/bonding-curves-in-depth-intuition-parametrization-d3905a681e0a.

[2] Slava Balasanov. *How to Make Bonding Curves for Continuous Token Models*. Feb. 6, 2018. URL: https://blog.relevant.community/how-to-make-bonding-curves-for-continuous-token-models-3784653f8b17.

[3] Veronica Coutts. *An introduction to bonding curves, shapes and use cases*. Aug. 2, 2019. URL: https://medium.com/linum-labs/intro-to-bonding-curves-and-shapes-bf326bc4e11a.

[4] Justin Goro. *Token Bonding Curves Explained*. Apr. 30, 2018. URL: https://medium.com/coinmonks/token-bonding-curves-explained-7a9332198e0e.

[5] Christopher H. Gorman. *Efficient Integer Square Roots in Solidity*. Oct. 2022. URL: https://github.com/alicenet/.github/blob/main/docs/efficient_isqrt.pdf.

[6] Eyal Hertzog, Guy Benartzi, and Galia Benartzi. *Bancor Protocol*. Whitepaper. Mar. 18, 2018. URL: https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf.

[7] Wilson Lau. *On Single Bonding Curves for Continuous Token Models*. July 5, 2018. URL: https://medium.com/thoughtchains/on-single-bonding-curves-for-continuous-token-models-a167f5ffef89.

[8] Yos Riady. *Bonding Curves Explained*. Nov. 10, 2018. URL: https://yos.io/2018/11/10/bonding-curves/.

[9] Simon de la Rouviere. *Bancor's Smart Tokens vs Token Bonding Curves*. Nov. 13, 2018. URL: https://medium.com/@simondlr/bancors-smart-tokens-vs-token-bonding-curves-a4f0cdfd3388.

[10] Simon de la Rouviere. *Tokens 2.0: Curved Token Bonding in Curation Markets*. Nov. 21, 2017. URL: https://medium.com/@simondlr/tokens-2-0-curved-token-bonding-in-curation-markets-1764a2e0bee5.

[11] Titian Steiger. *Advancing Decentralised Funding Mechanisms*. May 29, 2019. URL: https://medium.com/molecule-blog/designing-different-fundraising-scenarios-with-sigmoidal-token-bonding-curves-ceafc734ed97.

[12] Mikhail Vladimirov. *Math in Solidity (Part 1: Numbers)*. Feb. 17, 2020. URL: https://medium.com/coinmonks/math-in-solidity-part-1-numbers-384c8377f26d.

[13] Mikhail Vladimirov. *Math in Solidity (Part 5: Exponent and Logarithm)*. Apr. 2, 2020. URL: https://medium.com/coinmonks/math-in-solidity-part-5-exponent-and-logarithm-9aef8515136e.