Dynamic Automated Market Maker (DAMM)

Kamil Alizai Sadik

April 2021

Objective

The objective of this paper is to create an automated market maker (AMM) which:

1. Provides constant liquidity for an ERC-20 token on the Ethereum blockchain
2. Generates a profit for the issuer of the token

We will begin by reviewing existing approaches to providing liquidity and generating revenue using automated market matters to identify their strengths and shortcomings. We will then propose slight enhancements enabling a platform to generate revenue by providing liquidity. Lastly, we will propose a dynamic automated market maker to satisfy a precisely-defined profit objective.

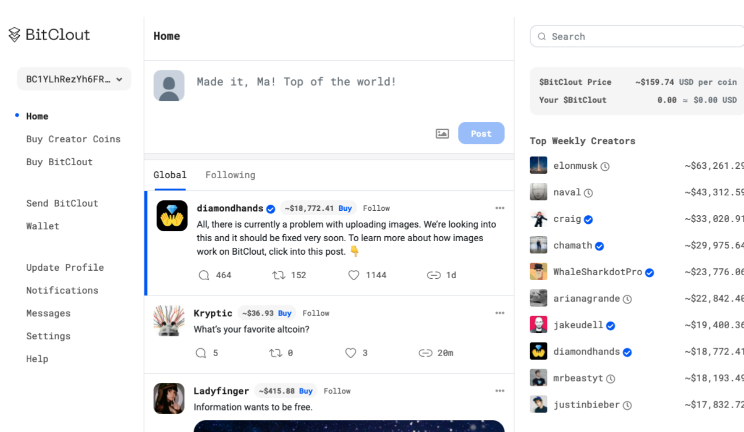
Existing Approaches

BitClout.com sells an intermediary currency (BitClout) as a revenue generator, and then provides zero-profit liquidity on tokens in the BitClout.com ecosystem (known as Creator Coins). The principal drawback of this approach is that there is no liquidity in the BitClout currency, resulting in secondary market transactions at a 40% discount on platforms like Discord.

Continuous Organizations uses no intermediary currency, and generates revenue through the sale of issued tokens, but with poor liquidity in the issued token.

BitClout

BitClout.com sells a currency called BitClout. Users are able to buy BitClout in exchange for Bitcoin. BitClout.com crystallizes its revenue by freely minting BitClout, and earning Bitcoin in exchange. Users buy BitClout since it is the only token with which they can purchase Creator Coins on the BitClout.com platform. At inception (last month), BitClout.com scraped the handles and avatars of the top 15,000 Twitter profiles, and created a Creator Coin for each of them. Their objective is to create a speculative, blockchain-based asset class out of social capital. Notably, BitClout.com makes liquid markets on each Creator Coin, but provides no liquidity on the BitClout currency itself, forcing users to resort to platforms like Discord to actually crystallize their BitClout profits.



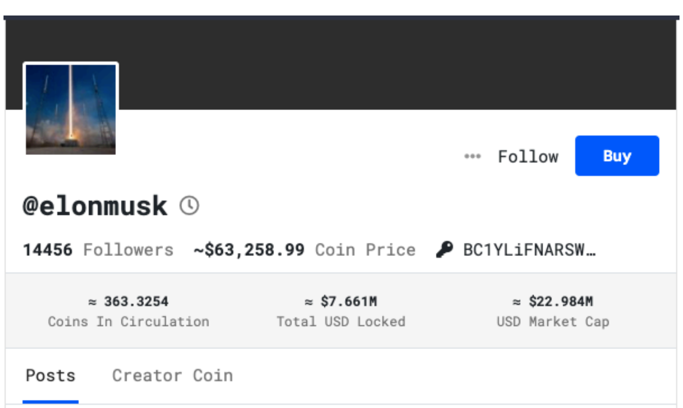
Source: BitClout.com

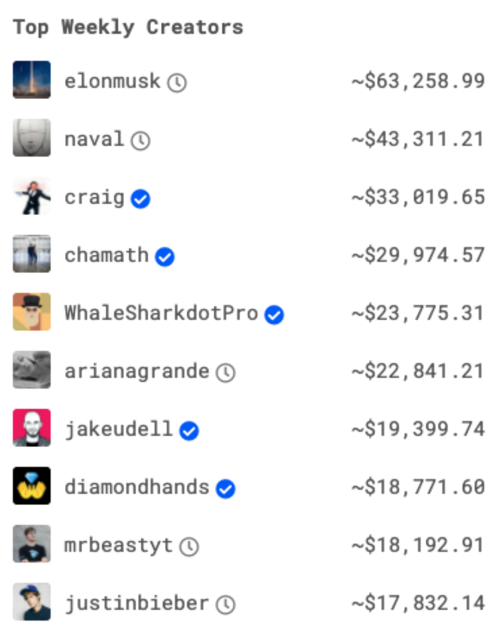
The revenue model of BitClout.com is predicated on the value of the BitClout currency. BitClout.com entices prospective customers into buying the BitClout currency in the hopes of realizing a profit by speculating on Creator Coins, and with the guarantee that the new issue price of BitClout will go up over time.

“The price of BitClout doubles for every million BitClout sold.”
— BitClout White Paper

Source: BitClout.com

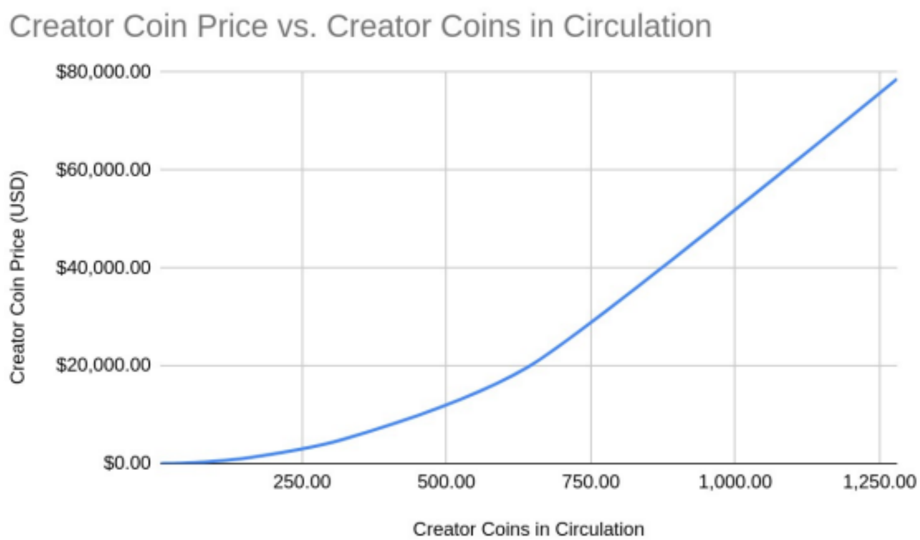
Users then use BitClout to purchase Creator Coins on the BitClout.com platform. BitClout.com also allows the creator referenced in each Creator Coin to claim a portion of the outstanding amount of their Creator Coin in exchange for verifying their account by tweeting that they are on BitClout.com (much like the DuoBlitz “promote to verify" strategy). Users can also create new Creator Coins of their own, and can select the portion of float that they would like to retain for themselves.





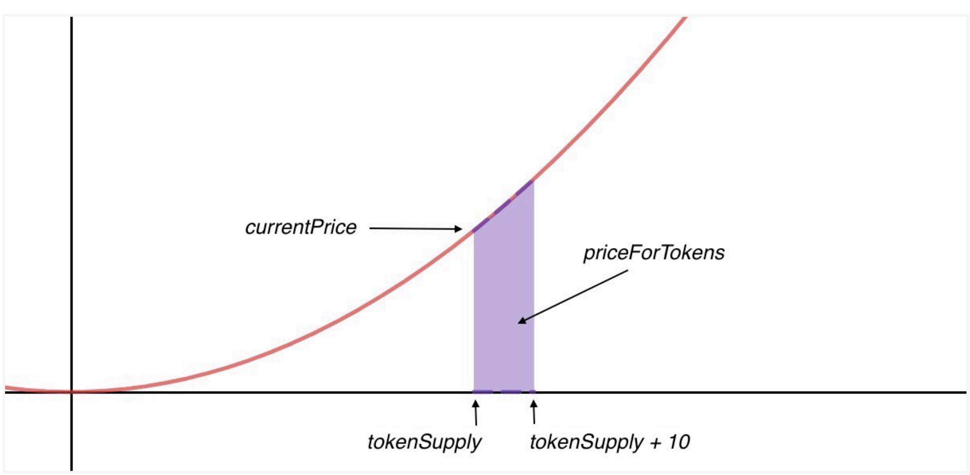
Source: BitClout.com

In order to provide liquidity on Creator Coins, BitClout.com uses an automated market maker (AMM), with the price of a Creator Coin determined by the quantity of that Coin in circulation. Automated market makers are a novel and increasingly popular approach to providing guaranteed liquidity in nascent markets. They have the advantage of constant liquidity, but the disadvantage that every trade moves the token's price (whereas, in a bilateral exchange, two large flows in opposite directions could have minimal price impact). As more users buy a Creator Coin, the price of the coin increases according to a fixed formula. The curve represents the price of the Creator Coin as a function of the quantity of that Creator Coin in circulation.



Source: BitClout White Paper

Buying increases the number of Creator Coins in circulation, and pushes the Creator Coin price up the curve. Selling decreases the number of Creator Coins in circulation, and pushes the Creator Coin price down the curve. The area under the curve, from 0 to the current price, represents the total value spent on that Creator Coin.



Source: Yos Riady, Bonding Curves Explained

Formally, given a Creator Coin with price curve $p(x)$, if there are k Creator Coins already in circulation, and a user submits an order to buy n Creator Coins:

- The number of Creator Coins in circulation after the transaction will be $k + n$
- The total cost of buying those n coins is $\int_{k}^{k+n} p(x)dx$
- The implied cost per coin purchased in the transaction is $\frac{\int_{k}^{k+n} p(x)dx}{n}$

The BitClout.com Creator Coin price function $p(x)$ is defined as $p(x) = 0.0003x^2 \cdot c$, where c is the cost of BitClout in USD, and x is the quantity of that Creator Coin in circulation.

D5 | fx | =$A$5*C5^2*A$2

	A	B	C	D	E
1	**BitClout Price in USD**	**USD in Profile**	**Coins sold**	**Coin price**	**Coin market cap**
2	1.00	$1	10.00	$0.30	$3.00
3	**Don't change**	$2	12.60	$0.48	$6.00
4	m	$4	15.87	$1.20 × $0.76	$12.00
5	0.003000	$8	20.00	=$A$5*C5^2*A$2	$24.00
6	r	$16	25.20	$1.90	$48.00
7	0.3333	$32	31.75	$3.02	$96.00
8		$64	40.00	$4.80	$192.00
9		$128	50.40	$7.62	$384.00
10		$256	63.50	$12.10	$768.00
11		$512	80.00	$19.20	$1,536.00
12		$1,024	100.79	$30.48	$3,072.00
13		$2,048	126.99	$48.38	$6,144.00
14		$4,096	160.00	$76.80	$12,288.00
15		$8,192	201.59	$121.91	$24,576.00
16		$16,384	253.98	$193.52	$49,152.00
17		$32,768	320.00	$307.20	$98,304.00
18		$65,536	403.17	$487.65	$196,608.00
19		$131,072	507.97	$774.10	$393,216.00
20		$262,144	640.00	$1,228.80	$786,432.00
21		$524,288	806.35	$1,950.60	$1,572,864.00
22		$1,048,576	1015.94	$3,096.38	$3,145,728.00
23		$2,097,152	1280.00	$4,915.20	$6,291,456.00
24		$4,194,304	1612.70	$7,802.39	$12,582,912.00
25		$8,388,608	2031.87	$12,385.53	$25,165,824.00
26		$16,777,216	2560.00	$19,660.80	$50,331,648.00
27		$33,554,432	3225.40	$31,209.57	$100,663,296.00
28		$67,108,864	4063.75	$49,542.11	$201,326,592.00

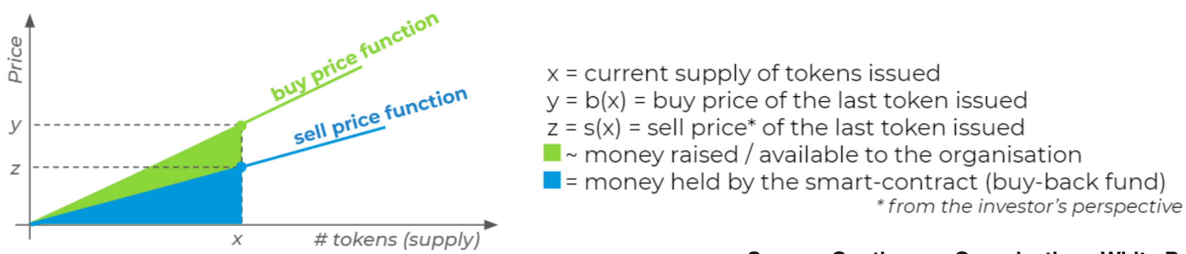
Source: BitClout Creator Coin Curve

BitClout.com does not earn any revenue through the sale of, or through making markets on, Creator Coins. All the revenue generated by the sale of Creator Coins functions as a liquidity pool, which stands ready to buy back the entire float of Creator Coins according to the price curve. The reason why the Creator Coin price curve generates no profit is because it allows user to both buy and sell the Creator Coin along the same price curve.

While BitClout.com has rapidly grown in popularity, with $225 million of BitClout purchases within one month of launch, there are some notable shortcomings. Firstly, having to convert Bitcoin to BitClout adds friction to the user experience. The fact that BitClout.com has generated $225 million in revenue in a single month (in spite of this friction) is a testament to the volume of cryptocurrency assets ready to be deployed into speculative opportunities. Secondly, BitClout.com offers no way to sell BitClout, forcing users who wish to take profit to do so on third-party platforms at a significant concession. BitClout.com creates the illusion of liquidity in Creator Coins, while simultaneously locking in profit through the sale of BitClout currency, for which the platform provides no liquidity.

Continuous Organizations

The Continuous Organizations approach was created as a way for startups to issue and provide automatic liquidity on tokens, while locking in capital with which to build the startup. It is an early paradigm for startups to raise capital in cryptocurrency markets. The Continuous Organizations approach uses a similar automated market maker to BitClout.com, with the notable difference that users buy the issued token along a steeper price curve, and sell the issued token along a less steep price curve.



Source: Continuous Organizations White Paper

The area under the sell price function constitutes the liquidity pool, which stands ready to buy tokens back. The area between the buy price function and the sell price function constitutes revenue locked in by the start-up. These price functions offer token-holders liquidity, but with substantial (and visible) bid offer, which grows as more tokens are issued. Notably, each time the supply of tokens moves up or down the price curves, the issuer captures significant bid offer.

The wide and increasing bid/offer associated with the Continuous Organizations approach is its chief weakness, and makes it undesirable for a platform which wishes to keep trade flows captive. Users are incentivized to trade on a third-party exchange to improve liquidity, which the Continuous Organizations white paper openly admits:

"Finally, it is good to keep in mind that the DAT is only the buyer-of-last-resort. It is very likely that an investor could sell their [tokens] on the secondary market at a higher price than the "buy-back price" offered by the cash reserve of the DAT for a given supply."
— Continuous Organizations White Paper

Dynamic Automated Market Maker (DAMM)

We wish to combine the advantages of the BitClout.com and Continuous Organizations approaches, while mitigating their shortcomings. Drawing from the Continuous Organizations approach, we wish to allow users to buy tokens issued by a platform (which we will refer to as in-network tokens) using Ethereum, without any intermediary currency. Drawing from BitClout.com's approach to Creator Coin market making, we wish to allow users to buy and sell at approximately the same level. We wish to accomplish both of these objectives on a platform which both generates and locks in revenue.

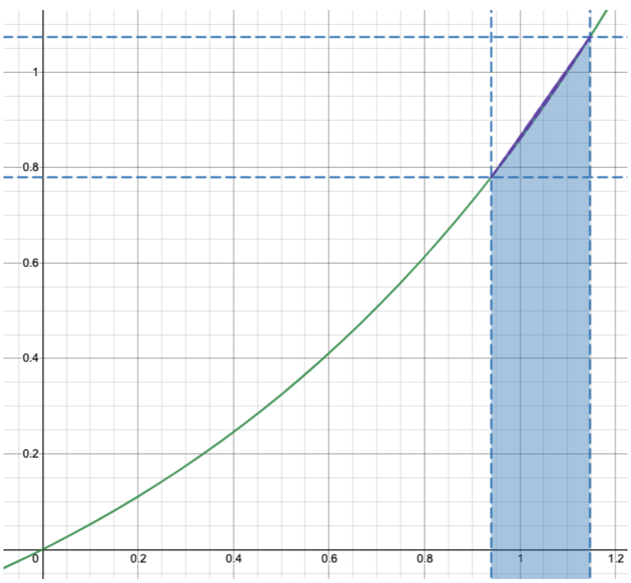
There are two approaches we can take in order to generate revenue while providing liquidity. The first is to extract bid/offer on each individual transaction. This enables the platform to generate revenue as trades take place. The second approach is to construct separate buy and sale price functions, with a separate sale price function for each level of supply in order to create a region of crystallized profit under the buy price curve.

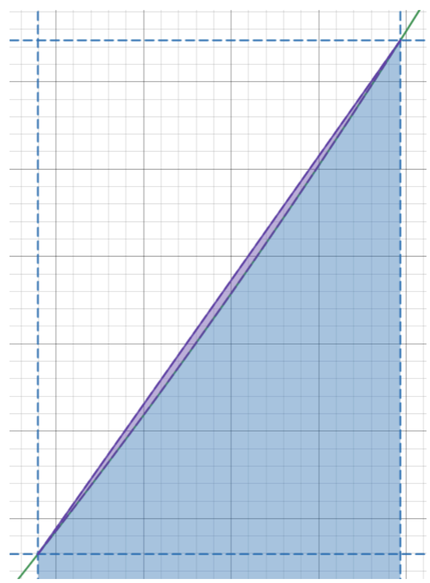
Bid/Offer Approach

Under the bid/offer approach, we will use a single price function for both buys and sales, with slight adjustments in order to extract bid/offer.

Straight-Line Bid/Offer

The first method of calculating bid/offer is to draw a straight-line between the point on the price curve corresponding to the supply of tokens before the transaction and the supply of tokens after the transaction.





Formally, given a price function $p(x)$, the bid/offer under this construct would be

$$\frac{(p(current\_supply) + p(post\_transaction\_supply)) \cdot (post\_transaction\_supply - current\_supply)}{2} - \int_{current\_supply}^{post\_transaction_supply} p(x)dx$$

Applying this approach, we find that bid/offer becomes extremely small, even for meaningful transaction sizes, as the price of the token increases.

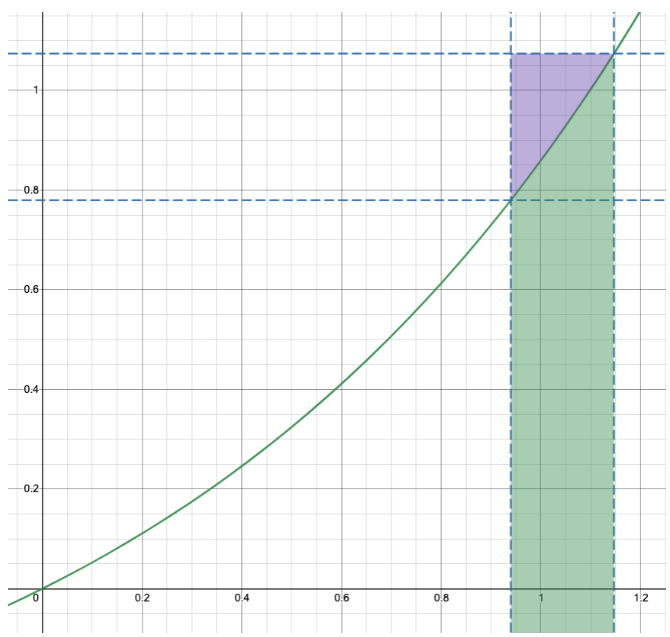
current_supply	current_price	post_transaction_supply	transaction_value	raw_transaction_price	bid_offer	bid_offer_percentage	gross_transaction_price	post_transaction_price	market_cap	total_revenue
0	$0.00	25	$15.63	$0.63	$7.812500	50.0000000%	$0.94	$1.88	$0.00	$0.00
0	$0.00	50	$125.00	$2.50	$62.500000	50.0000000%	$3.75	$7.50	$0.00	$0.00
50	$7.50	60	$91.00	$9.10	$0.500000	0.5494505%	$9.15	$10.80	$375.00	$125.00
100	$30.00	120	$728.00	$36.40	$4.000000	0.5494505%	$36.60	$43.20	$3,000.00	$1,000.00
150	$67.50	165	$1,117.13	$74.48	$1.687500	0.1510574%	$74.59	$81.68	$10,125.00	$3,375.00
200	$120.00	210	$1,261.00	$126.10	$0.500000	0.0396511%	$126.15	$132.30	$24,000.00	$8,000.00
400	$480.00	405	$2,430.13	$486.03	$0.062500	0.0025719%	$486.04	$492.08	$192,000.00	$64,000.00
700	$1,470.00	702	$2,948.41	$1,474.20	$0.004000	0.0001357%	$1,474.21	$1,478.41	$1,029,000.00	$343,000.00
1000	$3,000.00	1001	$3,003.00	$3,003.00	$0.000500	0.0000167%	$3,003.00	$3,006.00	$3,000,000.00	$1,000,000.00
2000	$12,000.00	2000.5	$6,001.50	$12,003.00	$0.000062	0.0000010%	$12,003.00	$12,006.00	$24,000,000.00	$8,000,000.00

Definitions

- current_supply refers to the number of tokens in circulation before the transaction
- current_price refers to the price of a token before the transaction
- post_transaction_supply refers to the number of tokens that will be in circulation after the transaction
- transaction_value refers to the area under the price curve between the current_supply and post_transaction_supply
- raw_transaction_price is the implied cost per token in the transaction, before the addition of bid/offer
- bid_offer refers to the total dollar amount of bid/offer paid on the transaction
- bid_offer_percentage expresses bid_offer as a percentage of transaction_value
- gross_transaction_price refers to the price per token traded in the transaction, inclusive of bid_offer
- post_transaction_price refers to the quoted price after the transaction is complete
- market_cap is the product of post_transaction_supply and post_transaction_price
- total_revenue is the area under the price curve from 0 to post_transaction_supply

Rectangular Bid/Offer

Given how quickly bid/offer on transactions tends towards zero as the price increases, it is tempting to attempt a more aggressive rectangular approach to bid/offer.



This approach actually performs even more poorly, since it punishes early transactions more aggressively, while still generating minimal bid/offer for later transactions.

current_supply	current_price	post_transaction_supply	transaction_value	raw_transaction_price	bid_offer	bid_offer_percentage	gross_transaction_price	post_transaction_price	market_cap	total_revenue
0	$0.00	25	$15.63	$0.63	$31.250000	200.0000000%	$1.88	$1.88	$0.00	$0.00
0	$0.00	50	$125.00	$2.50	$250.000000	200.0000000%	$7.50	$7.50	$0.00	$0.00
50	$7.50	60	$91.00	$9.10	$17.000000	18.6813187%	$10.80	$10.80	$375.00	$125.00
100	$30.00	120	$728.00	$36.40	$136.000000	18.6813187%	$43.20	$43.20	$3,000.00	$1,000.00
150	$67.50	165	$1,117.13	$74.48	$108.000000	9.6676737%	$81.68	$81.68	$10,125.00	$3,375.00
200	$120.00	210	$1,261.00	$126.10	$62.000000	4.9167328%	$132.30	$132.30	$24,000.00	$8,000.00
400	$480.00	405	$2,430.13	$486.03	$30.250000	1.2447919%	$492.08	$492.08	$192,000.00	$64,000.00
700	$1,470.00	702	$2,948.41	$1,474.20	$8.416000	0.2854422%	$1,478.41	$1,478.41	$1,029,000.00	$343,000.00
1000	$3,000.00	1001	$3,003.00	$3,003.00	$3.002000	0.0999667%	$3,006.00	$3,006.00	$3,000,000.00	$1,000,000.00
2000	$12,000.00	2000.5	$6,001.50	$12,003.00	$1.500250	0.0249979%	$12,006.00	$12,006.00	$24,000,000.00	$8,000,000.00

Fixed Commission

These results suggest that a preferable bid/offer construct is to simply charge a fixed percentage fee on all transactions, and to include that mark-up in the price at which the user executes their trade. Below is an example of a fixed 5% commission on all trades. This has the benefit of certainty of trading revenue without overcharging early trades and undercharging later trades.

current_supply	current_price	post_transaction_supply	transaction_value	raw_transaction_price	bid_offer	bid_offer_percentage	gross_transaction_price	post_transaction_price	market_cap	total_revenue
0	$0.00	25	$15.63	$0.63	$0.781250	5.0000000%	$0.66	$1.88	$0.00	$0.00
0	$0.00	50	$125.00	$2.50	$6.250000	5.0000000%	$2.63	$7.50	$0.00	$0.00
50	$7.50	60	$91.00	$9.10	$4.550000	5.0000000%	$9.56	$10.80	$375.00	$125.00
100	$30.00	120	$728.00	$36.40	$36.400000	5.0000000%	$38.22	$43.20	$3,000.00	$1,000.00
150	$67.50	165	$1,117.13	$74.48	$55.856250	5.0000000%	$78.20	$81.68	$10,125.00	$3,375.00
200	$120.00	210	$1,261.00	$126.10	$63.050000	5.0000000%	$132.41	$132.30	$24,000.00	$8,000.00
400	$480.00	405	$2,430.13	$486.03	$121.506250	5.0000000%	$510.33	$492.08	$192,000.00	$64,000.00
700	$1,470.00	702	$2,948.41	$1,474.20	$147.420400	5.0000000%	$1,547.91	$1,478.41	$1,029,000.00	$343,000.00
1000	$3,000.00	1001	$3,003.00	$3,003.00	$150.150050	5.0000000%	$3,153.15	$3,006.00	$3,000,000.00	$1,000,000.00
2000	$12,000.00	2000.5	$6,001.50	$12,003.00	$300.075006	5.0000000%	$12,603.15	$12,006.00	$24,000,000.00	$8,000,000.00

This approach works well for tokens issued by the platform, for which commission-based revenue might be sufficient; but it is less applicable when fundraising is the chief objective, as in the case of a user-created token for the purposes of raising money for an artistic project or startup (i.e., the use case of the Continuous Organizations approach). When fundraising is the objective, the token issuer would likely wish to retain a percentage of revenue generated through sale of the token which would be too large to charge as a transaction fee without undermining the goal of tight liquidity on the token. The token issuer would also likely wish to have their fundraising tied to total inflows of capital into their token rather than to trading volume in their token, particularly if token buyers opt to buy and hold rather than actively trading.

Dynamic Automated Market Maker (DAMM)

These shortcomings provide the motivation for a dynamic automated market maker. The DAMM allows users to buy in-network tokens using Ethereum, without any intermediary currency. Users can buy and sell tokens at approximately the same level, and the platform can generate and lock in revenue.

Like the Continuous Organizations approach, the DAMM uses separate buy and sale price functions. The critical difference in the DAMM, however, is that there are infinitely many sale price functions, each corresponding to a different level of peak supply that token has seen to date.

Definitions

- The max_supply is the highest recorded number of tokens in circulation to date, even if the current number of tokens in circulation is lower than max_supply
- The buy price function $b(x)$ is the function showing the impact of a buy which sets new highs (both in price, and in max_supply)
- The sale price function $s(x)$ is the function along which users both buy and sell when current token supply is less than max_supply
- Total revenue refers to the total proceeds generated through the sale of tokens, and is represented by the area under the buy price function from zero to max_supply. Formally,

$$total\_revenue = \int_0^{max\_supply} b(x)dx$$

- Profit refers to the portion of total revenue that is locked in, even in the event of a sell-off. It is represented by the area between the buy price function and sale price function from zero to max_supply. Formally,

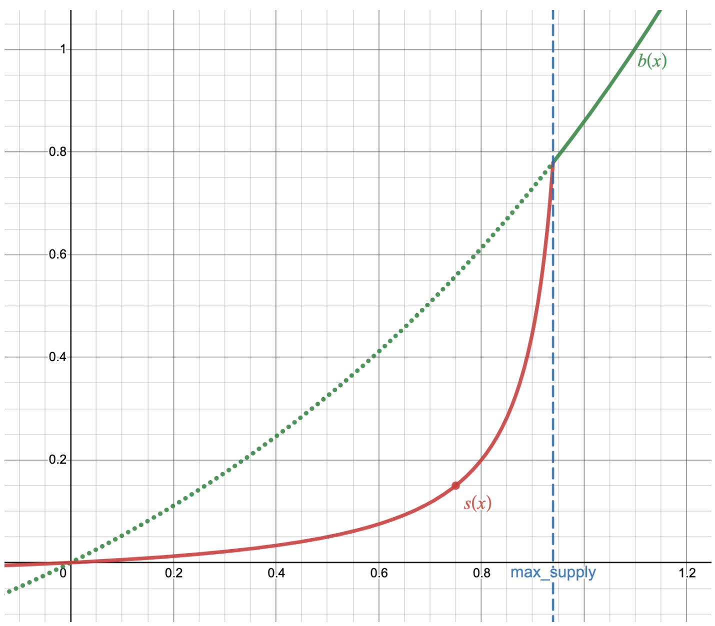
$$profit = \int_0^{max\_supply} b(x) - s(x)dx$$

- Profit margin refers to the percentage of total revenue that is crystallized profit. Formally,

$$profit\_margin = \frac{profit}{total\_revenue}$$

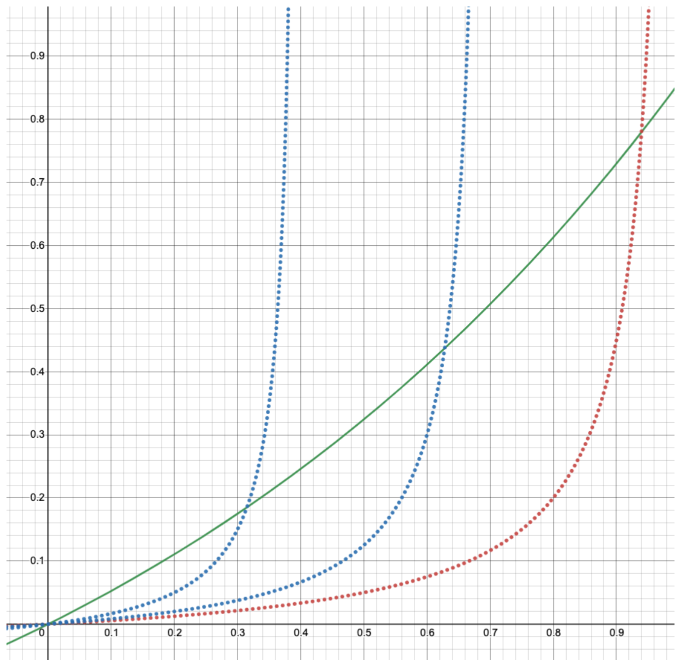
- Liquidity pool refers to the portion of total revenue that is held in reserve to provide liquidity on tokens. It is represented by the area below the sale price function from zero to max_supply. Formally,

$$liquidity\_pool = \int_0^{max\_supply} s(x)dx$$

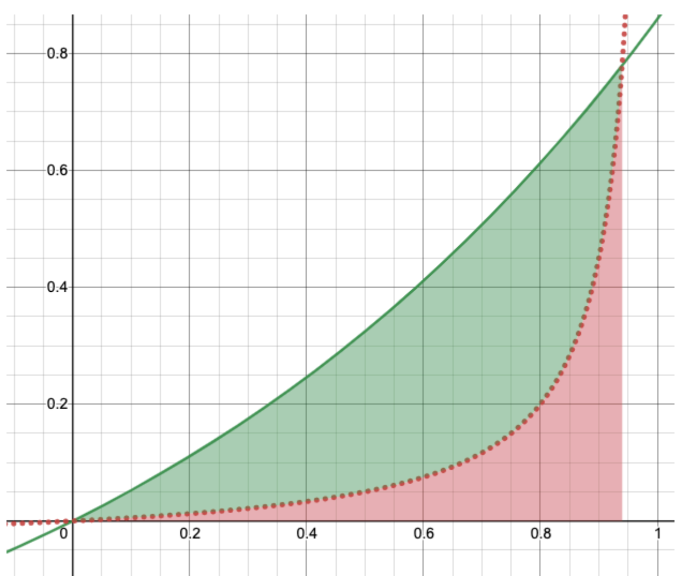


The DAMM uses separate buy and sale price functions, with a unique sale price function for each level of max_supply. We will more formally define the constraints for each of these functions later. For now, suffice it to say that $b(x)$ and $s(x)$ intersect at the origin, and at $(max\_supply, b(x))$. When the current number of tokens is equal to max_supply, each incremental buy moves the price up $b(x)$. When the current number of tokens is less than max_supply, both buys and sells take place along the sale price curve until max_supply reaches new highs.

While there are infinitely many sale price functions (one for each level of max_supply), it is important to note that, once a new level of max_supply is set, sale functions corresponding to lower levels of max_supply (shown in blue below) are never used again.



The key to the DAMM is that it locks in revenue by making sell-offs more aggressive than rallies. In the chart below, the green area between $b(x)$ and $s(x)$ is profit. This area grows as the token reaches new levels of max_supply, and is preserved in the event of a sell-off (since we use the sale price function $s(x)$ corresponding to the current level of max_supply). The red area under that sale price function is the liquidity pool, which stands ready to buy back the outstanding float of tokens.



While users are able to buy and sell at approximately the same price, there is a tradeoff between revenue and liquidity. If we optimize for a large profit area, there will be worse liquidity in the event of a sell-off, in the sense that a trade of a given size will have a larger impact on the token's price than would an identically sized trade with a smaller (or zero) profit area.

Since one of the principal motivations for the DAMM is fundraising, it stands to reason that we will want to pre-define our desired profit margin, and to construct buy and sale price functions which solve for that profit margin. We will thus select a buy function $b(x)$, and define a set of sale price functions (one for each level of max_supply) which solve for our chosen profit margin, while also satisfying the more general liquidity-providing objectives.

Having established the intuition behind the DAMM, we now establish the formalized constraints for our buy function $b(x)$ and sale function $s(x)$.

Given a desired profit margin, chosen such that $0 \leq profit\_margin \leq 1$.

Define a buy price function $b(x)$ such that:

1. $b(0) = 0$
2. b is monotonically nondecreasing on $[0, \infty)$
3. The slope of b is monotonic on $[0, \infty)$

For each $max\_supply \: \epsilon \: [0, \infty)$, define a sale price function $s(x)$ such that:

4. $s(0) = 0$
5. s is monotonically nondecreasing on $[0, max\_supply]$
6. $s(max\_supply) = b(max\_supply)$
7. $\dfrac{\int_0^{max\_supply} s(x)dx}{\int_0^{max\_supply} b(x)dx} = 1 - profit\_margin$
8. $s(x) \leq b(x) \quad \forall x \: \epsilon \: [0, max\_supply]$

Constraints 1 and 4 ensure that the buy and sale price functions both meet at the origin so that there is smooth liquidity at inception, and in the event of a sell-off resulting in a token's price going to zero.

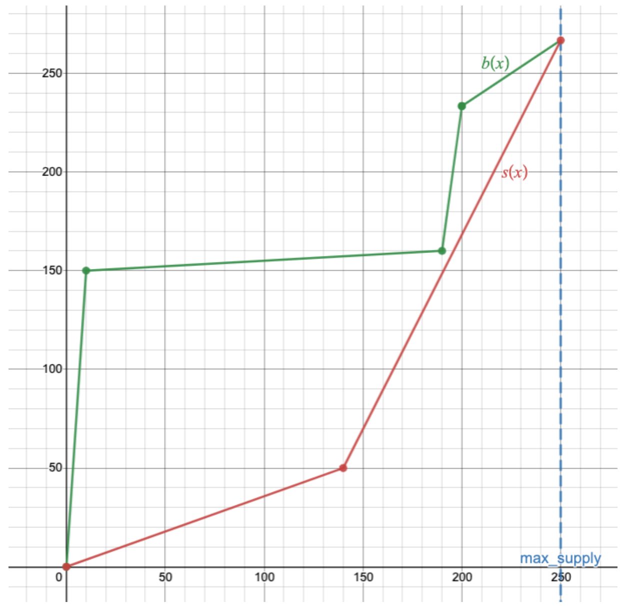
Constraint 6 is similar in motivation to Constraints 1 and 4, in that it serves to guarantee smooth liquidity when the current supply of tokens is at or near the current max_supply level.

Constraints 2 and 5 ensure that the price of the token increases as more tokens are minted. Otherwise, it would create undesirable incentives if users were buying along a curve which would diminish the value of their tokens with additional purchases.

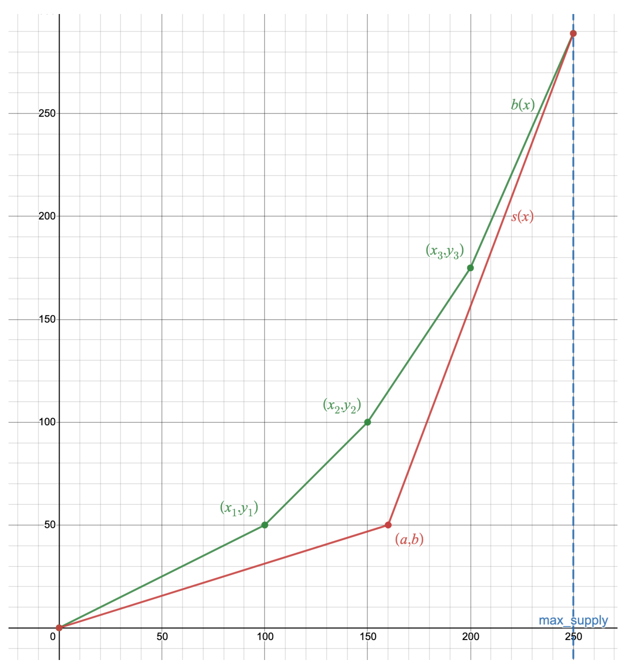
Constraint 7 forces us to construct buy and sale functions such that we solve for our desired profit margin. Intuitively, the area under sale function $s(x)$ should be $(1 - profit\_margin)$% of the area under buy function $b(x)$.

Constraint 8 guarantees that at no point in the course of a sell-off are tokens trading at a price exceeding the price at which they would have traded during a rally with the same supply of tokens.

Lastly, Constraint 3 is a heuristic to ensure that we avoid a bizarrely constructed buy function which might result in an excessive profit margin. Consider the extreme example below showing piecewise-defined buy and sale functions. This buy function results either in a very large profit margin, or requires additional inflection points on $s(x)$, creating unnecessary complexity in our definition of $s(x)$.



Having established these constraints, we can develop an intuition for what our buy and sale price functions ought to look like. For simplicity, we use piecewise-defined linear buy and sale price functions, noting that we can approximate any curve of our choosing with a sufficiently granular buy function $b(x)$.

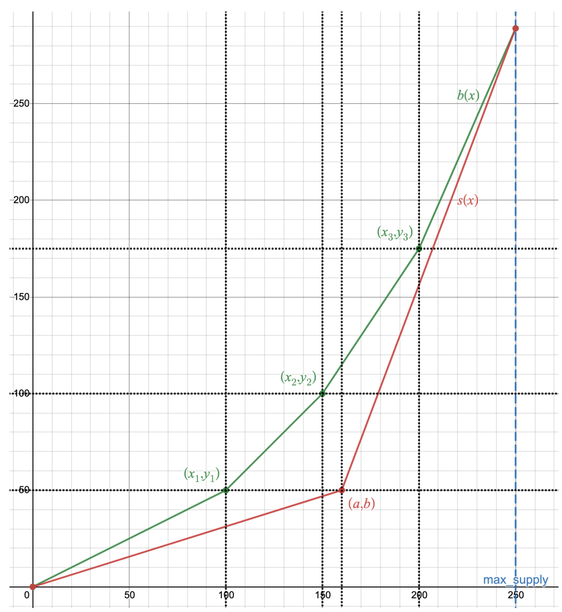


For our buy price function, we allow for multiple breakpoints (non-differentiable points on $b(x)$ where the slope changes abruptly), notating these n-many breakpoints as (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n), where $x_k \epsilon [0, max\_supply]$ $\forall k \epsilon \{1,...,n\}$

For our sale price function $s(x)$, we allow for a single breakpoint (a, b), where $a \epsilon [0, max\_supply]$. As we will discover, even this single breakpoint on $s(x)$ allows for sufficient flexibility in our final selection of sale price function.

Using this notation, we can now formalize our expression of our buy function $b(x)$, and the set of corresponding sale functions $s(x)$:

> Given a piecewise-defined buy function $b(x)$, consisting of linear segments, and constructed to satisfy Constraints 1, 2, and 3; and given some arbitrary $max\_supply \geq 0$, we wish to derive a piecewise-defined sale function $s(x)$, consisting of linear segments, and satisfying Constraints 4, 5, 6, 7, and 8.
>
> We define a breakpoint to be a non-differentiable point on an otherwise differentiable piecewise-defined function. We construct buy function $b(x)$ such that it may have n-many breakpoints (as defined above). Similarly, we wish to derive sale function $s(x)$ such that it has a single breakpoint (a, b).



> We begin by applying Constraint 7, finding the area under the buy and sale price functions by breaking these areas into triangles and trapezoids.

$$\frac{\int_0^{max\_supply} s(x)dx}{\int_0^{max\_supply} b(x)dx} = 1 - profit\_margin$$

Substituting into Constraint 7, we get:

$$\frac{\frac{a \cdot b}{2} + \frac{(b + b(max\_supply))(max\_supply - a)}{2}}{\frac{x_1 \cdot y_1}{2} + \frac{(y_1 + y_2)(x_2 - x_1)}{2} + \frac{(y_2 + y_3)(x_3 - x_2)}{2} + \ldots + \frac{(b(\max\_supply) + y_n)(max\_supply - x_n)}{2}} = 1 - profit\_margin$$

Simplifying gives:

$$a \cdot b + (b + b(max\_supply))(max\_supply - a) = (1 - profit\_margin) \cdot (x_1 \cdot y_1 + (y_1 + y_2)(x_2 - x_1) + (y_2 + y_3)(x_3 - x_2) + \ldots + (b(\max\_supply) + y_n)(max\_supply - x_n))$$

Finally, we rearrange this equation to isolate a and b to the extent possible:

$$b \cdot max\_supply - a \cdot b(max\_supply) = (1 - profit\_margin) \cdot (x_1 \cdot y_1 + (y_1 + y_2)(x_2 - x_1) + (y_2 + y_3)(x_3 - x_2) + \ldots + (b(\max\_supply) + y_n)(max\_supply - x_n)) - max\_supply \cdot b(max\_supply)$$

This equation gives us a range of candidate points (a, b) to choose from as a function of our construction of $b(x)$ and our desired profit margin. We may opt to select relatively close values of a and b to help ensure that we satisfy Constraint 8, and to smooth out liquidity providing such that liquidity is acceptable both near max_supply and during a protracted sell-off.

By construction, we have satisfied Constraints 1-6. We will be careful to select (a, b) such that we satisfy Constraint 8. By the derivation above, we have identified the set of points (a, b) which result in a sale price function satisfying Constraint 7.

Use Cases

The two broad use cases for any of these liquidity providing frameworks are platform-created tokens and user-created tokens.

Platform-created tokens

Platform-created tokens are minted and issued by a for-profit trading platform. If using a DAMM, the platform would keep the crystallized profit associated with the issuance of tokens, and would provide liquidity along the buy and sale price functions as described above.

A fixed commission approach with a single price function for both buys and sales is also a valid approach for a for-profit platform. The choice between a DAMM and a fixed commission approach is a tradeoff between prioritizing rapid revenue capture (in the case of a DAMM), and better liquidity over time (in the case of a modest fixed commission).

An example of such a for-profit platform is a Fantasy Football trading platform. The platform would issue a token representing each NFL player, and users would buy tokens as a way of speculating on that player's performance, and the impact of that performance on the token's value.

Intuitively, it makes sense to lean towards a fixed commission approach when targeting a market likely to trade large volumes actively, and to lean towards the DAMM when targeting a market more likely to buy and hold.

User-created tokens

User-created tokens are minted and issued on a platform by a user of that platform. Users can create these tokens in order to generate profit for themselves by utilizing a DAMM and keeping the crystallized profit region for themselves. Since the combination of a DAMM and a meaningful fixed commission would unduly hamper liquidity, the platform could extract revenue by keeping a portion of the DAMM's profit region, or by charging only a nominal fixed commission on trades.

Examples of user-created tokens include startups issuing tokens as an analogy to issuing shares in fiat currency markets; and artists issuing tokens to fund a work of art, which also already takes place in the fiat currency market (e.g., Protest the Hero's crowdfunded album, *Volition*). Artists in particular can offer tangible non-financial value to their token holders by offering perks such as monthly Zoom calls, backstage meet and greets, and other exclusive benefits. Fans may also buy the token of a little-known artist whose popularity they expect to increase over time, and benefit financially as that artist becomes more successful. Artists may opt to offer perks based on volume of token ownership (e.g., their top 10 token holders) as well as by duration of token ownership (e.g., the 25 longest holders of that artist's token). The potential for duration-based rewards is further motivation for the DAMM over a fixed commission, since duration-based rewards will have a tendency to reduce trading volume.

These use cases are particularly important, in that they intermediate between cryptocurrency holders and individuals who need and have a productive use for capital. Given the volume of value which lives in cryptocurrency markets, an efficient capital allocation scheme, delivering capital to those who need it, and yield to those who have it, is very much in order.