A General Theory of Liquidity Provisioning for Automated Market Makers

Adithya Bhaskara
University of Colorado Boulder
adithya@colorado.edu

July 8, 2024

Work supported by the National Science Foundation and Ethereum Foundation

Collaborators

Rafael Frongillo

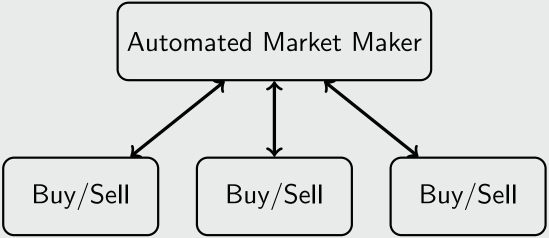
Maneesha Papireddygari

A, Rafael Frongillo, and Maneesha Papireddygari. A general theory of liquidity provisioning for automated market makers, 2023. URL https://arxiv.org/abs/2311.08725.

raf@colorado.edu, maneesha.papireddygari@colorado.edu

1. Motivation

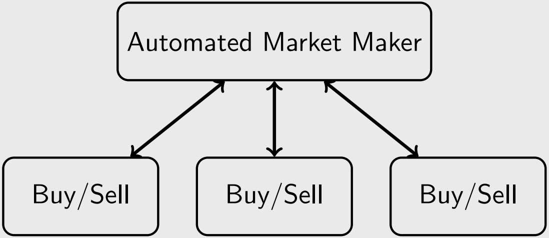
Automated Market Makers



Hanson [2003] introduced automated market makers (AMMs) to solve thin market problems in prediction markets.

AMMs are always willing to offer a price for any bundle of assets.

Automated Market Makers

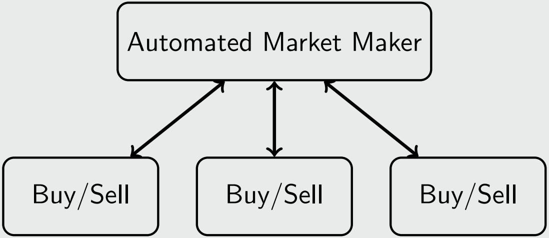


Hanson [2003] introduced automated market makers (AMMs) to solve thin market problems in prediction markets.

AMMs are always willing to offer a price for any bundle of assets.

AMMs ideally ensure that prices are stable, and we quantify the degree of price stability with the term liquidity.

Automated Market Makers



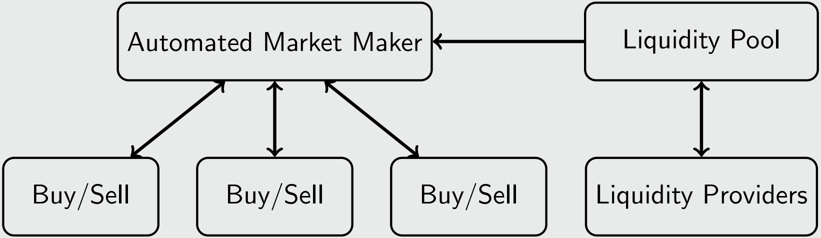
Hanson [2003] introduced automated market makers (AMMs) to solve thin market problems in prediction markets.

AMMs are always willing to offer a price for any bundle of assets.

AMMs ideally ensure that prices are stable, and we quantify the degree of price stability with the term liquidity.

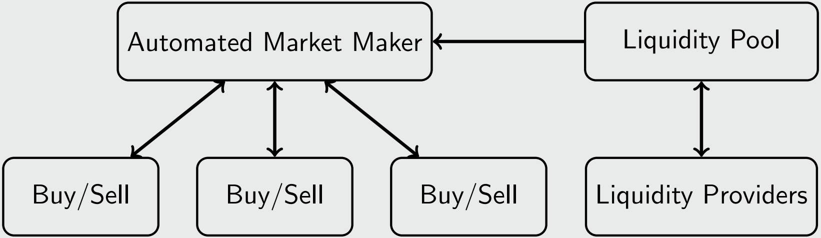
Traditionally, market creators both fostered the trade of assets and bore the risk of providing liquidity to keep prices stable.

Liquidity Provisioning



Recently, AMMs have been used as a easy way to trade assets on chain in decentralized finance (DeFi), [Bartoletti et al., 2022, Xu et al., 2023, Mohan, 2022].

Liquidity Provisioning



Recently, AMMs have been used as a easy way to trade assets on chain in decentralized finance (DeFi), [Bartoletti et al., 2022, Xu et al., 2023, Mohan, 2022].

We see a decoupling of roles: the market creator may outsource liquidity provison to external liquidity providers (LPs).

LPs are compensated for taking on risk by getting a cut of fees skimmed off trades.

The Design of Liquidity Provisioning Protocols

Where does the current state of the design space of liquidity provision protocols lie?

The Design of Liquidity Provisioning Protocols

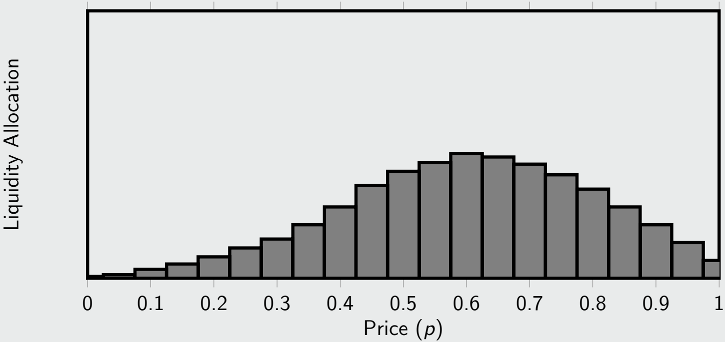
Where does the current state of the design space of liquidity provision protocols lie?

Consider Uniswap V3, where LPs allocate liquidity by depositing assets in various discrete "buckets" that partition the price space, while traders exchange assets.

The Design of Liquidity Provisioning Protocols

Where does the current state of the design space of liquidity provision protocols lie?

Consider Uniswap V3, where LPs allocate liquidity by depositing assets in various discrete "buckets" that partition the price space, while traders exchange assets.



The Design of Liquidity Provisioning Protocols

How do we implement Uniswap V3?

How do we implement Uniswap V3?

function ModifyLiquidity($i \in \mathbb{N}, \alpha' \geq 0, j \in \{0, \ldots, m\}$)

  $p = \text{price}(\mathbf{x})$

  request $\mathbf{x}' = \begin{cases} \left((\alpha' - \alpha^{ij})\left(\sqrt{\frac{1-a_j}{a_j}} - \sqrt{\frac{1-b_j}{b_j}}\right), 0\right) & \text{if } p < a_j \\ \left(0, (\alpha' - \alpha^{ij})\left(\sqrt{\frac{b_j}{1-b_j}} - \sqrt{\frac{a_j}{1-a_j}}\right)\right) & \text{if } p > b_j \\ \left((\alpha' - \alpha^{ij})\left(\sqrt{\frac{1-p}{p}} - \sqrt{\frac{1-b_j}{b_j}}\right), (\alpha' - \alpha^{ij})\left(\sqrt{\frac{p}{1-p}} - \sqrt{\frac{a_j}{1-a_j}}\right)\right) & \text{if } p \in [a_j, b_j] \end{cases}$

  $(\mathbf{x}, \alpha^{ij}) \leftarrow (\mathbf{x} + \mathbf{x}', \alpha')$

function ExecuteTrade($\mathbf{r} \in \mathbb{R}^2$)

  Let $p = \text{price}(\mathbf{x})$, $p' = \text{price}(\mathbf{x} - \mathbf{r})$

  Let l, u be such that $a_l \leq p \leq b_l$ and $a_u \leq p' \leq b_u$.

  check

$$\frac{1}{(\sum_{i=0}^k \alpha^{il})^2}\left(x_1 + \sum_{i=0}^k \alpha^{il}\sqrt{\frac{1-b_l}{b_l}}\right)\left(x_2 + \sqrt{\frac{a_l}{1-a_l}}\sum_{i=0}^k \alpha^{il}\right) = \frac{1}{(\sum_{i=0}^k \alpha^{iu})^2}\left(x_1 - r_1 + \sum_{i=0}^k \alpha^{iu}\sqrt{\frac{1-b_u}{b_u}}\right)\left(x_2 - r_2 + \sqrt{\frac{a_u}{1-a_u}}\sum_{i=0}^k \alpha^{iu}\right)$$

  pay $\beta \frac{\sum_j \alpha^{ij}}{\sum_j \sum_o \alpha^{oj}}(-\mathbf{r})_+$ to each LP i where j sums over buckets in $[B^l, B^u]$.

  $\triangleright$ WLOG the B^u bucket comes later than the B^l bucket.

  $\mathbf{x} \leftarrow \mathbf{x} - \mathbf{r}$

The Design of Liquidity Provisioning Protocols

How do we implement Uniswap V3?

function ModifyLiquidity($i \in \mathbb{N}, \alpha' \geq 0, j \in \{0, \ldots, m\}$)

  $p = \text{price}(\mathbf{x})$

  request $\mathbf{x}' = \begin{cases} \left((\alpha' - \alpha^{ij})\left(\sqrt{\frac{1-a_j}{a_j}} - \sqrt{\frac{1-b_j}{b_j}}\right), 0\right) & \text{if } p < a_j \\ \left(0, (\alpha' - \alpha^{ij})\left(\sqrt{\frac{b_j}{1-b_j}} - \sqrt{\frac{a_j}{1-a_j}}\right)\right) & \text{if } p > b_j \\ \left((\alpha' - \alpha^{ij})\left(\sqrt{\frac{1-p}{p}} - \sqrt{\frac{1-b_j}{b_j}}\right), (\alpha' - \alpha^{ij})\left(\sqrt{\frac{p}{1-p}} - \sqrt{\frac{a_j}{1-a_j}}\right)\right) & \text{if } p \in [a_j, b_j] \end{cases}$

  $(\mathbf{x}, \alpha^{ij}) \leftarrow (\mathbf{x} + \mathbf{x}', \alpha')$

function ExecuteTrade($\mathbf{r} \in \mathbb{R}^2$)

  Let $p = \text{price}(\mathbf{x})$, $p' = \text{price}(\mathbf{x} - \mathbf{r})$

  Let l, u be such that $a_l \leq p \leq b_l$ and $a_u \leq p' \leq b_u$.

  check

$$\frac{1}{(\sum_{i=0}^{k} \alpha^{il})^2}\left(x_1 + \sum_{i=0}^{k} \alpha^{il}\sqrt{\frac{1-b_l}{b_l}}\right)\left(x_2 + \sqrt{\frac{a_l}{1-a_l}}\sum_{i=0}^{k} \alpha^{il}\right) = \frac{1}{(\sum_{i=0}^{k} \alpha^{iu})^2}\left(x_1 - r_1 + \sum_{i=0}^{k} \alpha^{iu}\sqrt{\frac{1-b_u}{b_u}}\right)\left(x_2 - r_2 + \sqrt{\frac{a_u}{1-a_u}}\sum_{i=0}^{k} \alpha^{iu}\right)$$

  pay $\beta \frac{\sum_j \alpha^{ij}}{\sum_j \sum_o \alpha^{oj}}(-\mathbf{r})_+$ to each LP i where j sums over buckets in $[B^l, B^u]$.

  ▷ WLOG the B^u bucket comes later than the B^l bucket.

  $\mathbf{x} \leftarrow \mathbf{x} - \mathbf{r}$

Well, . . . , that's a bit scary.

Expanding the Design Space of Liquidity Provisioning Protocols

Questions

Expanding the Design Space of Liquidity Provisioning Protocols

Questions

1. *What is a principled way to come up with liquidity provisioning protocols like this?*

Expanding the Design Space of Liquidity Provisioning Protocols

Questions

1. *What is a principled way to come up with liquidity provisioning protocols like this?*
2. *How can we characterize the full design space of liquidity provisioning protocols?*

Expanding the Design Space of Liquidity Provisioning Protocols

Questions

1. *What is a principled way to come up with liquidity provisioning protocols like this?*
2. *How can we characterize the full design space of liquidity provisioning protocols?*

Answer (Our Framework)

Think about liquidity provisioning as LPs running their own automated market makers operating in parallel.

The Rest of This Talk

1. Our New Framework
2. Examples
3. Recap & Future Directions

2. Our New Framework

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Question: What would running CFMMs $\varphi^1, \ldots, \varphi^n$ in parallel look like?

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Question: What would running CFMMs $\varphi^1, \ldots, \varphi^n$ in parallel look like?
Claim: A CFMM with $\varphi = \bigwedge_i \varphi^i$, where $\bigwedge$ is the *infimal convolution* can be thought of as doing this.

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Question: What would running CFMMs $\varphi^1, \ldots, \varphi^n$ in parallel look like?
Claim: A CFMM with $\varphi = \bigwedge_i \varphi^i$, where $\bigwedge$ is the *infimal convolution* can be thought of as doing this.

Our Framework

1. Think about CFMMs in parallel. What does that mean?

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Question: What would running CFMMs $\varphi^1, \ldots, \varphi^n$ in parallel look like?
Claim: A CFMM with $\varphi = \bigwedge_i \varphi^i$, where $\bigwedge$ is the *infimal convolution* can be thought of as doing this.

Our Framework

1. Think about CFMMs in parallel. What does that mean?
2. Use the answer to create a general, maximally flexible LP protocol.

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Question: What would running CFMMs $\varphi^1, \dots, \varphi^n$ in parallel look like?
Claim: A CFMM with $\varphi = \bigwedge_i \varphi^i$, where $\bigwedge$ is the *infimal convolution* can be thought of as doing this.

Our Framework

1. Think about CFMMs in parallel. What does that mean?
2. Use the answer to create a general, maximally flexible LP protocol.
3. Characterize the design space of LP protocols.

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Question: What would running CFMMs $\varphi^1, \dots, \varphi^n$ in parallel look like?
Claim: A CFMM with $\varphi = \bigwedge_i \varphi^i$, where $\bigwedge$ is the *infimal convolution* can be thought of as doing this.

Our Framework

1. Think about CFMMs in parallel. What does that mean?
2. Use the answer to create a general, maximally flexible LP protocol.
3. Characterize the design space of LP protocols.

Thm 1. We can recover all existing protocols, like Uniswap V2 and V3.

Overview of the Framework

In traditional markets, liquidity is provided by many market makers quoting prices in parallel.

This narrows the bid-ask spread, leading to price stability.

Question: What would running CFMMs $\varphi^1, \ldots, \varphi^n$ in parallel look like?
Claim: A CFMM with $\varphi = \bigwedge_i \varphi^i$, where $\bigwedge$ is the *infimal convolution* can be thought of as doing this.

Our Framework

1. Think about CFMMs in parallel. What does that mean?
2. Use the answer to create a general, maximally flexible LP protocol.
3. Characterize the design space of LP protocols.

Thm 1. We can recover all existing protocols, like Uniswap V2 and V3.
Thm 2. The general protocol is "reasonable" for any "reasonable" φ^i.

Background: Constant Function Market Makers

A CFMM maintains a reserve vector $\boldsymbol{x} \in \mathbb{R}^n$ and a convex[1] potential function[2] $\varphi : \mathbb{R}^n \to \mathbb{R}$. The trades $\boldsymbol{r} \in \mathbb{R}^n$ available are those satisfying $\varphi(\boldsymbol{x} + \boldsymbol{r}) = \varphi(\boldsymbol{x})$. After a trade, $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{r}$.

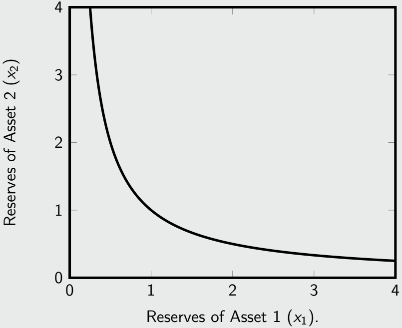
[1] Conventionally, φ is concave; we use the negation as the potential.

[2] For readers familiar with the prediction market literature, [Chen and Pennock, 2007, Abernethy et al., 2013, Frongillo et al., 2023], φ and φ^i as used in our slides are really cost functions.

Background: Constant Function Market Makers

A CFMM maintains a reserve vector $\boldsymbol{x} \in \mathbb{R}^n$ and a convex[1] potential function[2] $\varphi : \mathbb{R}^n \to \mathbb{R}$. The trades $\boldsymbol{r} \in \mathbb{R}^n$ available are those satisfying $\varphi(\boldsymbol{x} + \boldsymbol{r}) = \varphi(\boldsymbol{x})$. After a trade, $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{r}$.

For example, take Uniswap V2 with $\varphi(\boldsymbol{x}) = x_1 x_2$:



[1] Conventionally, φ is concave; we use the negation as the potential.

[2] For readers familiar with the prediction market literature, [Chen and Pennock, 2007, Abernethy et al., 2013, Frongillo et al., 2023], φ and φ^i as used in our slides are really cost functions.

Main Results

Definition (Infimal Convolution)

For convex functions $f^i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$*, we define their infimal convolution* $f = \bigwedge_i f^i$ *by* $f(\boldsymbol{x}) = \inf \left\{ \sum_i f^i(\boldsymbol{x}^i) \mid \sum_i \boldsymbol{x}^i = \boldsymbol{x} \right\}$*, where the* $\boldsymbol{x}^i$ *range over* $\mathbb{R}^n$*.*

Alternatively, $\wedge$ *is the conjugate operator of* $+$*.*

Main Results

Definition (Infimal Convolution)

For convex functions $f^i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, *we define their infimal convolution* $f = \bigwedge_i f^i$ *by* $f(\boldsymbol{x}) = \inf \left\{ \sum_i f^i(\boldsymbol{x}^i) \mid \sum_i \boldsymbol{x}^i = \boldsymbol{x} \right\}$, *where the* $\boldsymbol{x}^i$ *range over* $\mathbb{R}^n$.

Alternatively, $\wedge$ *is the conjugate operator of* $+$.

Theorem 1 (Existing LP Protocols → Aggregate CFMM)

All existing liquidity provisioning protocols correspond to an aggregate CFMM with potential $\varphi = \bigwedge \varphi^i$ *for certain choices of* φ^i, *where* φ^i *is the CFMM that LP* i *chooses.*

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions?

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\rightarrow$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \ldots, \varphi^n$ corresponds to a reasonable LP protocol.

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\rightarrow$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \ldots, \varphi^n$ corresponds to a reasonable LP protocol.

Maximally Flexible LP Protocol From Our Framework

1. Each LP chooses and modifies their own CFMM φ^i arbitrarily.

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\rightarrow$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \dots, \varphi^n$ corresponds to a reasonable LP protocol.

Maximally Flexible LP Protocol From Our Framework

1. Each LP chooses and modifies their own CFMM φ^i arbitrarily.
   AMM requests assets such that aggregate reserves are nonnegative.

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\rightarrow$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \ldots, \varphi^n$ corresponds to a reasonable LP protocol.

Maximally Flexible LP Protocol From Our Framework

1. Each LP chooses and modifies their own CFMM φ^i arbitrarily.
   AMM requests assets such that aggregate reserves are nonnegative.
2. The aggregate market maker uses $\varphi = \bigwedge_i \varphi^i$ to emulate the LPs' choices of CFMMs operating in parallel.

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\rightarrow$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \ldots, \varphi^n$ corresponds to a reasonable LP protocol.

Maximally Flexible LP Protocol From Our Framework

1. Each LP chooses and modifies their own CFMM φ^i arbitrarily.
   AMM requests assets such that aggregate reserves are nonnegative.
2. The aggregate market maker uses $\varphi = \bigwedge_i \varphi^i$ to emulate the LPs' choices of CFMMs operating in parallel.
3. Valid trades $\boldsymbol{r}$ are those satisfying $\varphi(\boldsymbol{x} + \boldsymbol{r}) = \varphi(\boldsymbol{x})$.

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\rightarrow$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \ldots, \varphi^n$ corresponds to a reasonable LP protocol.

Maximally Flexible LP Protocol From Our Framework

1. Each LP chooses and modifies their own CFMM φ^i arbitrarily.
   AMM requests assets such that aggregate reserves are nonnegative.
2. The aggregate market maker uses $\varphi = \bigwedge_i \varphi^i$ to emulate the LPs' choices of CFMMs operating in parallel.
3. Valid trades $\boldsymbol{r}$ are those satisfying $\varphi(\boldsymbol{x} + \boldsymbol{r}) = \varphi(\boldsymbol{x})$.

Reasonable?

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\to$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \ldots, \varphi^n$ corresponds to a reasonable LP protocol.

Maximally Flexible LP Protocol From Our Framework

1. Each LP chooses and modifies their own CFMM φ^i arbitrarily.
   AMM requests assets such that aggregate reserves are nonnegative.
2. The aggregate market maker uses $\varphi = \bigwedge_i \varphi^i$ to emulate the LPs' choices of CFMMs operating in parallel.
3. Valid trades $\boldsymbol{r}$ are those satisfying $\varphi(\boldsymbol{x} + \boldsymbol{r}) = \varphi(\boldsymbol{x})$.

Reasonable? Running CFMMs in parallel, ✓.

Using our Framework to Create New Protocols

In existing protocols like Uniswap, LPs are constrained to a certain class of φ^i. Can we generalize this to a broader class of functions? Yes!

Theorem 2 (Aggregate CFMM $\rightarrow$ Reasonable LP Protocol)

Any choice of convex $\varphi^1, \ldots, \varphi^n$ corresponds to a reasonable LP protocol.

Maximally Flexible LP Protocol From Our Framework

1. Each LP chooses and modifies their own CFMM φ^i arbitrarily.
   AMM requests assets such that aggregate reserves are nonnegative.
2. The aggregate market maker uses $\varphi = \bigwedge_i \varphi^i$ to emulate the LPs' choices of CFMMs operating in parallel.
3. Valid trades $\boldsymbol{r}$ are those satisfying $\varphi(\boldsymbol{x} + \boldsymbol{r}) = \varphi(\boldsymbol{x})$.

Reasonable? Running CFMMs in parallel, $\checkmark$. Liquidity adds, $\checkmark$.

Sense Check 1: Running CFMMs in Parallel

Another[3], more "literal," translation of running CFMMs in parallel:

[3] 2 other reasonable implementations of what it means for CFMMs to run in parallel, mentioned in our paper, are equivalent!

Sense Check 1: Running CFMMs in Parallel

Another[3], more "literal," translation of running CFMMs in parallel:

- Let CFMM i have potential function φ^i. If traders want to make a net trade $\boldsymbol{r}$, they do so by making trades $\boldsymbol{r}^i$ with each CFMM φ^i.

[3] 2 other reasonable implementations of what it means for CFMMs to run in parallel, mentioned in our paper, are equivalent!

Sense Check 1: Running CFMMs in Parallel

Another[3], more "literal," translation of running CFMMs in parallel:

- Let CFMM i have potential function φ^i. If traders want to make a net trade $\boldsymbol{r}$, they do so by making trades $\boldsymbol{r}^i$ with each CFMM φ^i.
- Rational traders wish to trade at the best prices offered.

[3] 2 other reasonable implementations of what it means for CFMMs to run in parallel, mentioned in our paper, are equivalent!

Sense Check 1: Running CFMMs in Parallel

Another[3], more "literal," translation of running CFMMs in parallel:

- Let CFMM i have potential function φ^i. If traders want to make a net trade $\boldsymbol{r}$, they do so by making trades $\boldsymbol{r}^i$ with each CFMM φ^i.
- Rational traders wish to trade at the best prices offered.

Trading with the aggregate CFMM $\varphi = \bigwedge_i \varphi^i$ does exactly that!

[3] 2 other reasonable implementations of what it means for CFMMs to run in parallel, mentioned in our paper, are equivalent!

Sense Check 1: Running CFMMs in Parallel

Another[3], more "literal," translation of running CFMMs in parallel:

- Let CFMM i have potential function φ^i. If traders want to make a net trade $\boldsymbol{r}$, they do so by making trades $\boldsymbol{r}^i$ with each CFMM φ^i.
- Rational traders wish to trade at the best prices offered.

Trading with the aggregate CFMM $\varphi = \bigwedge_i \varphi^i$ does exactly that!

As an added benefit, computing the optimal trade split $\boldsymbol{r}$ under the hood allows traders not needing to choose which CFMMs φ^i to trade with for best prices.

[3] 2 other reasonable implementations of what it means for CFMMs to run in parallel, mentioned in our paper, are equivalent!

Sense Check 2: Liquidity Adds!

Let $\varphi = \bigwedge_i \varphi^i$ be our convex aggregate potential. For intuition, suppose $\nabla^2 \varphi \succ \mathbf{0}$.

Sense Check 2: Liquidity Adds!

Let $\varphi = \bigwedge_i \varphi^i$ be our convex aggregate potential. For intuition, suppose $\nabla^2\varphi \succ \mathbf{0}$.

Intuitively, liquidity is a measure of how insensitive prices are while trades occur. At a price $\boldsymbol{p} = \nabla\varphi(\boldsymbol{x}) \in \Delta_n$, we can define $\ell(\boldsymbol{p}) = \left(\nabla^2\varphi\right)^{-1}(\boldsymbol{x})$.

Sense Check 2: Liquidity Adds!

Let $\varphi = \bigwedge_i \varphi^i$ be our convex aggregate potential. For intuition, suppose $\nabla^2 \varphi \succ \mathbf{0}$.

Intuitively, liquidity is a measure of how insensitive prices are while trades occur. At a price $\boldsymbol{p} = \nabla \varphi(\boldsymbol{x}) \in \Delta_n$, we can define $\ell(\boldsymbol{p}) = \left(\nabla^2 \varphi\right)^{-1}(\boldsymbol{x})$.

Because ℓ is positive definite, we can say $\ell = \nabla^2 \psi$ for some convex $\psi : \Delta_n \to \mathbb{R}$. This is a special case of convex conjugate duality; $\psi = \varphi^*$.

Sense Check 2: Liquidity Adds!

Let $\varphi = \bigwedge_i \varphi^i$ be our convex aggregate potential. For intuition, suppose $\nabla^2 \varphi \succ \mathbf{0}$.

Intuitively, liquidity is a measure of how insensitive prices are while trades occur. At a price $\boldsymbol{p} = \nabla \varphi(\boldsymbol{x}) \in \Delta_n$, we can define $\ell(\boldsymbol{p}) = \left(\nabla^2 \varphi\right)^{-1}(\boldsymbol{x})$.

Because ℓ is positive definite, we can say $\ell = \nabla^2 \psi$ for some convex $\psi : \Delta_n \to \mathbb{R}$. This is a special case of convex conjugate duality; $\psi = \varphi^*$.

By results in convex analysis, $\varphi^* = \left(\bigwedge_i \varphi^i\right)^* = \sum_i (\varphi^i)^*$.

Sense Check 2: Liquidity Adds!

Let $\varphi = \bigwedge_i \varphi^i$ be our convex aggregate potential. For intuition, suppose $\nabla^2 \varphi \succ \mathbf{0}$.

Intuitively, liquidity is a measure of how insensitive prices are while trades occur. At a price $\boldsymbol{p} = \nabla \varphi(\boldsymbol{x}) \in \Delta_n$, we can define $\ell(\boldsymbol{p}) = \left(\nabla^2 \varphi\right)^{-1}(\boldsymbol{x})$.

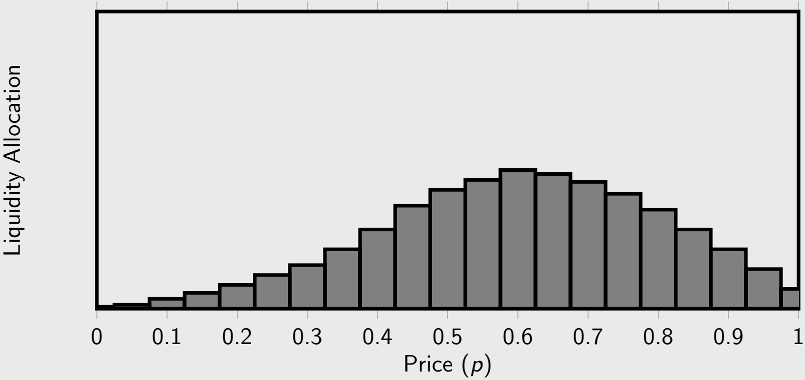
Because ℓ is positive definite, we can say $\ell = \nabla^2 \psi$ for some convex $\psi : \Delta_n \to \mathbb{R}$. This is a special case of convex conjugate duality; $\psi = \varphi^*$.

By results in convex analysis, $\varphi^* = \left(\bigwedge_i \varphi^i\right)^* = \sum_i (\varphi^i)^*$.

So, with $\ell = (\nabla^2 \varphi)^{-1} = \nabla^2 \varphi^*$, the total liquidity is the *sum* of the liquidity from each LP: $\ell = \sum_i \ell^i$.

3. Examples

Remember Uniswap V3's Buckets?



Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

[4]See Abernethy et al. [2013], Frongillo et al. [2023].

Generalizing the Idea of Buckets From Uniswap V3

Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

If an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$, how much of each asset would they have to provide if the current price was p?

[4] See Abernethy et al. [2013], Frongillo et al. [2023].

Generalizing the Idea of Buckets From Uniswap V3

Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

If an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$, how much of each asset would they have to provide if the current price was p?

	Uniswap V3	LMSR	Brier
$p \in [a_j, b_j]$	$\begin{pmatrix} \sqrt{\frac{1-b_j}{b_j}} - \sqrt{\frac{1-p}{p}} \\ \sqrt{\frac{a_j}{1-a_j}} - \sqrt{\frac{p}{1-p}} \end{pmatrix}$		

[4] See Abernethy et al. [2013], Frongillo et al. [2023].

Generalizing the Idea of Buckets From Uniswap V3

Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

If an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$, how much of each asset would they have to provide if the current price was p?

	Uniswap V3	LMSR	Brier
$p \in [a_j, b_j]$	$\begin{pmatrix} \sqrt{\frac{1-b_j}{b_j}} - \sqrt{\frac{1-p}{p}} \\ \sqrt{\frac{a_j}{1-a_j}} - \sqrt{\frac{p}{1-p}} \end{pmatrix}$	$\begin{pmatrix} \log \frac{p}{b_j} \\ \log \frac{1-p}{1-a_j} \end{pmatrix}$	

[4] See Abernethy et al. [2013], Frongillo et al. [2023].

Generalizing the Idea of Buckets From Uniswap V3

Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

If an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$, how much of each asset would they have to provide if the current price was p?

	Uniswap V3	LMSR	Brier
$p \in [a_j, b_j]$	$\begin{pmatrix} \sqrt{\frac{1-b_j}{b_j}} - \sqrt{\frac{1-p}{p}} \\ \sqrt{\frac{a_j}{1-a_j}} - \sqrt{\frac{p}{1-p}} \end{pmatrix}$	$\begin{pmatrix} \log \frac{p}{b_j} \\ \log \frac{1-p}{1-a_j} \end{pmatrix}$	$\begin{pmatrix} (1-b_j)^2 - (1-p)^2 \\ a_j^2 - p^2 \end{pmatrix}$

[4] See Abernethy et al. [2013], Frongillo et al. [2023].

Generalizing the Idea of Buckets From Uniswap V3

Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

If an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$, how much of each asset would they have to provide if the current price was p?

	Uniswap V3	LMSR	Brier
$p \in [a_j, b_j]$	$\begin{pmatrix} \sqrt{\frac{1-b_j}{b_j}} - \sqrt{\frac{1-p}{p}} \\ \sqrt{\frac{a_j}{1-a_j}} - \sqrt{\frac{p}{1-p}} \end{pmatrix}$	$\begin{pmatrix} \log \frac{p}{b_j} \\ \log \frac{1-p}{1-a_j} \end{pmatrix}$	$\begin{pmatrix} (1-b_j)^2 - (1-p)^2 \\ a_j^2 - p^2 \end{pmatrix}$

How did we get this?

[4] See Abernethy et al. [2013], Frongillo et al. [2023].

Generalizing the Idea of Buckets From Uniswap V3

Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

If an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$, how much of each asset would they have to provide if the current price was p?

	Uniswap V3	LMSR	Brier
$p \in [a_j, b_j]$	$\begin{pmatrix} \sqrt{\frac{1-b_j}{b_j}} - \sqrt{\frac{1-p}{p}} \\ \sqrt{\frac{a_j}{1-a_j}} - \sqrt{\frac{p}{1-p}} \end{pmatrix}$	$\begin{pmatrix} \log \frac{p}{b_j} \\ \log \frac{1-p}{1-a_j} \end{pmatrix}$	$\begin{pmatrix} (1-b_j)^2 - (1-p)^2 \\ a_j^2 - p^2 \end{pmatrix}$

How did we get this? Some technical details[4] involving the duality between reserve curves and scoring rules.

[4]See Abernethy et al. [2013], Frongillo et al. [2023].

Generalizing the Idea of Buckets From Uniswap V3

Let's say we wanted to use the bucketing mechanism from Uniswap, but with any choice of reserve curve. What's stopping us?

If an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$, how much of each asset would they have to provide if the current price was p?

	Uniswap V3	LMSR	Brier
$p \in [a_j, b_j]$	$\begin{pmatrix} \sqrt{\frac{1-b_j}{b_j}} - \sqrt{\frac{1-p}{p}} \\ \sqrt{\frac{a_j}{1-a_j}} - \sqrt{\frac{p}{1-p}} \end{pmatrix}$	$\begin{pmatrix} \log \frac{p}{b_j} \\ \log \frac{1-p}{1-a_j} \end{pmatrix}$	$\begin{pmatrix} (1-b_j)^2 - (1-p)^2 \\ a_j^2 - p^2 \end{pmatrix}$

How did we get this? Some technical details[4] involving the duality between reserve curves and scoring rules.

We can do the same for an arbitrary reserve curve!

[4] See Abernethy et al. [2013], Frongillo et al. [2023].

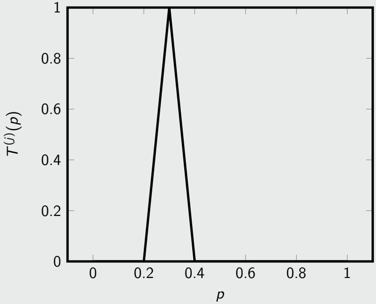
Triangular Liquidity

Bucketing requires LPs to discretize their liquidity allocations. For more expressiveness, let's try "soft" buckets that "fade" in and out around a target price a_j.

Triangular Liquidity

Bucketing requires LPs to discretize their liquidity allocations. For more expressiveness, let's try "soft" buckets that "fade" in and out around a target price a_j.

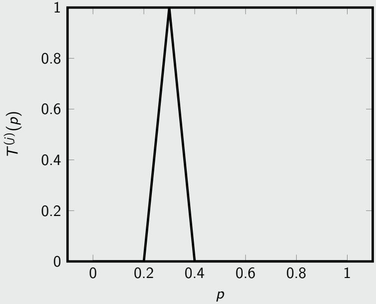
Consider a triangular function, $T^{(j)}(p)$.



Triangular Liquidity

Bucketing requires LPs to discretize their liquidity allocations. For more expressiveness, let's try "soft" buckets that "fade" in and out around a target price a_j.

Consider a triangular function, $T^{(j)}(p)$.



We can "weight" liquidity functions over "soft" bucket j with $\ell^{(j)}(p) = (\ell T^{(j)})(p)$.

Polynomial Liquidity

Bucketing also requires us to keep track of several parameters.

Polynomial Liquidity

Bucketing also requires us to keep track of several parameters.

Operations involving polynomials are efficient, and polynomials can approximate most functions.

Polynomial Liquidity

Bucketing also requires us to keep track of several parameters.

Operations involving polynomials are efficient, and polynomials can approximate most functions.

Requiring LPs to choose polynomial liquidity functions ℓ of degree k allows expressiveness like bucketing, but only requires k parameters to manage.

Polynomial Liquidity

Bucketing also requires us to keep track of several parameters.

Operations involving polynomials are efficient, and polynomials can approximate most functions.

Requiring LPs to choose polynomial liquidity functions ℓ of degree k allows expressiveness like bucketing, but only requires k parameters to manage.

This could be especially promising in off-chain settings where the cost of inverting polynomials isn't an issue.

4. Recap & Future Directions

Recap & Future Directions

1. Recap: Thinking about LPs running CFMMs operating in parallel gives us a nice way to think about liquidity provisioning.

Recap & Future Directions

1. Recap: Thinking about LPs running CFMMs operating in parallel gives us a nice way to think about liquidity provisioning.

2. Recap: Within this framework, we can both understand existing protocols more deeply, and we can capture all conceivable protocols.

Recap & Future Directions

1. Recap: Thinking about LPs running CFMMs operating in parallel gives us a nice way to think about liquidity provisioning.

2. Recap: Within this framework, we can both understand existing protocols more deeply, and we can capture all conceivable protocols.

3. Future: Wrestle with hiccups that arise from more than 2 assets; issues with directional liquidity, [Grugett, 2023], and trading fees may not be budget balanced.

Recap & Future Directions

1. Recap: Thinking about LPs running CFMMs operating in parallel gives us a nice way to think about liquidity provisioning.
2. Recap: Within this framework, we can both understand existing protocols more deeply, and we can capture all conceivable protocols.
3. Future: Wrestle with hiccups that arise from more than 2 assets; issues with directional liquidity, [Grugett, 2023], and trading fees may not be budget balanced.
4. Future: Use this framework to create new protocols with considerations like Just-in-Time behavior, impermanent loss, and other desiderata in mind. We'd love to hear your suggestions!

Recap & Future Directions

Thanks!

1. Recap: Thinking about LPs running CFMMs operating in parallel gives us a nice way to think about liquidity provisioning.

2. Recap: Within this framework, we can both understand existing protocols more deeply, and we can capture all conceivable protocols.

3. Future: Wrestle with hiccups that arise from more than 2 assets; issues with directional liquidity, [Grugett, 2023], and trading fees may not be budget balanced.

4. Future: Use this framework to create new protocols with considerations like Just-in-Time behavior, impermanent loss, and other desiderata in mind. We'd love to hear your suggestions!

`adithya@colorado.edu`, `raf@colorado.edu`, `maneesha.papireddygari@colorado.edu`

References

Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation*, 1(2):12, 2013. URL http://dl.acm.org/citation.cfm?id=2465777.

Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch-Lafuente. A theory of automated market makers in defi. *Logical Methods in Computer Science*, Volume 18, Issue 4, 2022. ISSN 1860-5974. doi: 10.46298/lmcs-18(4:12)2022. URL http://dx.doi.org/10.46298/lmcs-18(4:12)2022.

Y. Chen and D.M. Pennock. A utility framework for bounded-loss market makers. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 49–56, 2007.

Rafael Frongillo, Maneesha Papireddygari, and Bo Waggoner. An axiomatic characterization of cfmms and equivalence to prediction markets, 2023. URL https://arxiv.org/abs/2302.00196.

James Grugett. Multiple choice markets, 2023. URL https://news.manifold.markets/p/multiple-choice-markets. Retrieved 11/14/2023.

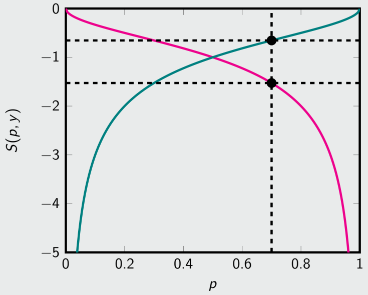
R. Hanson. Combinatorial Information Market Design. *Information Systems Frontiers*, 5(1):107–119, 2003.

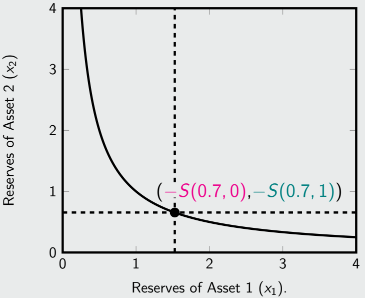
Vijay Mohan. Automated market makers and decentralized exchanges: a defi primer. *Financial Innovation*, 8(20), 2022. doi: 10.1186/s40854-021-00314-5. URL http://dx.doi.org/10.1145/3570639.

Jiahua Xu, Krzysztof Paruch, Simon Cousaert, and Yebo Feng. Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols. *ACM Computing Surveys*, 55(11):1–50, February 2023. ISSN 1557-7341. doi: 10.1145/3570639. URL http://dx.doi.org/10.1145/3570639.

In Case of Questions

Scoring Rule & Reserve Curve Duality





Buckets From Uniswap V3 With Any Scoring Rule

In general, if an LP wanted to deposit liquidity in a certain bucket $[a_j, b_j]$ in a CFMM using a scoring rule $S(p, y)$, they'd need to deposit

	General Scoring Rule $S(p, y)$
$p < a_j$	$\begin{pmatrix} S(a_j, 1) - S(b_j, 1) \\ 0 \end{pmatrix}$
$p \in [a_j, b_j]$	$\begin{pmatrix} S(p, 1) - S(b_j, 1) \\ S(p, 0) - S(a_j, 0) \end{pmatrix}$
$p > b_j$	$\begin{pmatrix} 0 \\ S(b_j, 0) - S(a_j, 0) \end{pmatrix}$