Batch Exchanges with Constant Function Market Makers: Axioms, Equilibria, and Computation

GEOFFREY RAMSEYER, Stanford University, USA
MOHAK GOYAL, Stanford University, USA
ASHISH GOEL, Stanford University, USA
DAVID MAZIÈRES, Stanford University, USA

Batch trading systems and constant function market makers (CFMMs) are two distinct market design innovations that have recently come to prominence as ways to address some of the shortcomings of decentralized trading systems. However, different deployments have chosen substantially different methods for integrating the two innovations.

We show here, from a minimal set of axioms describing the beneficial properties of each innovation, that there is only one unique method for integrating CFMMs to provide liquidity in batch trading schemes that preserves all the beneficial properties of both. Deployment of a batch trading scheme trading in many assets simultaneously requires a reliable algorithm for finding (approximate) equilibria in Arrow-Debreu exchange markets. We study this problem when batches contain limit orders and CFMMs. Specifically, we find that CFMM design affects the asymptotic complexity of the problem, give an easily-checkable criterion to validate that a given CFMM is computationally tractable in a batch trading system, and give a convex program that computes equilibria on batches of limit orders and CFMMs. This convex program computes equilibria of Arrow-Debreu exchange markets when every agent’s demand response satisfies weak gross substitutability, and every agent has utility for only two assets. This convex program has rational solutions when running on many (but not all) natural classes of widely-deployed CFMMs.

This work is supported by the Future of Digital Currency Initiative (FDCI), Stanford University, and the Office of Naval Research, award number N000141912268.

Authors’ addresses: Geoffrey Ramseyer, Stanford University, Stanford, California, 94305, USA, geoff.ramseyer@cs.stanford.edu; Mohak Goyal, Stanford University, Stanford, California, 94305, USA, mohakg@stanford.edu; Ashish Goel, Stanford University, Stanford, California, 94305, USA, ashishg@stanford.edu; David Mazières, Stanford University, Stanford, California, 94305, USA.
1 INTRODUCTION
A crucial component of an economic system is a structure to facilitate the exchange of assets. Typical exchanges facilitate the trading of asset pairs through continuous double auctions. Traders submit trade offers to an exchange, which either matches the new offer with an existing, compatible offer or, if none exists, adds the new offer to its orderbooks. Each offer has a limit price, and will accept a trade that gives an exchange rate at least as favorable as the limit price.

Because continuous double auctions process trades sequentially (in continuous time), every arbitrage opportunity or advantageous price fluctuation is captured by the party whose trade reaches the exchange first. The result is continual capital investment in ultra low-latency computer systems that, for those not participating in the arbitrage races, functions as a tax on liquidity provision [11, 16].

Budish et al. [16] propose using batch auctions to address these challenges. Trades between two assets are accumulated over a short period of time, after which the exchange operator computes a uniform clearing price and settles as many trades as possible. Batch exchange systems provide additional benefits for decentralized exchanges running in modern blockchains, specifically improving system scalability [47] and eliminating many kinds of widespread ordering manipulation attacks (e.g. [23]).

Several batch exchange systems are deployed or in development [1, 4, 6]. The NYSE schedules trades in batches at the opening and closing of a trading day [5]. These systems often model a batch of trades as an instance of an Arrow-Debreu exchange market [13]. In this model, each “agent” has an initial endowment of goods and a utility function that expresses preferences over sets of goods. The “market” posts a set of asset valuations (quotients of which give a set of exchange rates), and each agent independently computes an optimal (utility maximizing) trade with the market, trading at the posted valuations. We call an optimal trade an agent’s “demand response” to a set of posted valuations. An equilibrium of an exchange market consists of a set of valuations and a demand response for each agent; operating a batch exchange using this model, therefore, requires efficient computation of Arrow-Debreu exchange market equilibria.

Using this framework lets batch exchanges handle not just trading between pairs of assets (as in traditional markets) but also between batches of many assets. This can help users efficiently trade directly from one asset to any other asset (without holding an intermediate asset, such as USD). Furthermore, many-asset trading reduces liquidity fragmentation between asset pairs, a problem especially problematic in the modern blockchain ecosystem [43]. However, the computation of an exchange market equilibrium is more difficult (asymptotically and empirically) in many-asset exchange markets than in 2-asset exchange markets. Furthermore, the computational difficulty can depend greatly on the behavior of batch participants.

Independently, so-called “Constant Function Market Makers” (CFMMs) [8] have gained great prominence in the world of decentralized finance as a type of automated market-maker. Liquidity providers place capital, in the form of some units of two (or more) different assets, into the CFMM’s reserves. The CFMM offers then to trade these assets with external users according to a predefined trading strategy. Specifically, a CFMM is equipped with an eponymous “trading function” (a real-valued function of the quantities of the assets in its reserves), and accepts any trade that does not decrease the value of this function. CFMMs by nature require less active management than many market-making strategies, and can reduce the frequency of (expensive) transactions in many throughput-limited blockchain contexts. CFMMs, such as Uniswap [7] and Curve [28] are already some of the largest decentralized trading platforms.

Our topic of study here is on how these two market design innovations interact. How can CFMMs provide liquidity within a batch trading framework without disrupting the beneficial properties of the batch exchange? First, at minimum,
to fit into a batch exchange model, a CFMM needs a notion of a demand response, and ideally, a CFMM’s demand responses should be induced by a natural utility function. And second, to implement a batch exchange, an operator needs an efficient algorithm for computing batch solutions when some participants are (derived from) CFMMs. For batch exchanges modeled as Arrow-Debreu exchange markets, this means efficiently computing market equilibria.

However, there are, a priori, several natural choices for a CFMM’s demand response. In fact, different batch exchange deployments have chosen different demand responses, and academic literature has studied (analogues of) several more. Figure 1 depicts several possible choices. For example, the Stellar blockchain plans to use rule C [40], and CoWSwap uses rule B. Angeris et al. [9] study an analogue of rule A, and Milionis et al. [44] study (a continuous-time analogue of) rules E and F.

1.1 Our Results

We give (in §3) a minimal set of axioms describing batch trading schemes and CFMMs, and from these axioms show that trading rules C and D of Figure 1 are equivalent and are the unique, optimal ways to operate CFMMs within batch trading systems while preserving the beneficial properties of both.

![Fig. 1. Some plausible CFMM demand responses. Horizontal and vertical axes respectively denote the amount of assets $A$ and $B$ in the CFMM’s reserves. The solid blue curve denotes the level set of the CFMM’s eponymous trading function; Axiom 5 rules out trades to points below this curve. The slope of the dotted orange line denotes the set of trades at the batch exchange rate; Axiom rules out trades to points above this line. Some possible trading rules in the remaining space (formal descriptions deferred to §): Rule A) Trade until the CFMM’s spot exchange rate equals the batch exchange rate, keeping the CFMM trading function constant. Rule B) Trade as much as possible at the batch exchange rate, without causing a net decrease in the CFMM trading function. Rule C) Trade at the batch exchange rate until the CFMM’s spot exchange rate equals the batch exchange rate. Rule D) Trade at the batch exchange rate to maximize the CFMM trading function. Rule E) Follow rule A, but instead trade at the batch exchange rate, keeping surplus value in $A$. Rule F) Follow rule A, but instead trade at the batch exchange rate, keeping surplus value in $B$. Furthermore, these axioms lead to a complete specification for the equilibrium computation problem in batch trading schemes that include CFMMs and traditional limit orders. Specifically, we show (in §5) that a CFMM following “the optimal trading rule” can be naturally modeled as an agent within the Arrow-Debreu exchange market model, using the CFMM’s trading function as an Arrow-Debreu agent’s utility function.
Arrow-Debreu exchange markets admit polynomial-time equilibria computation algorithms when agent demand responses satisfy a property known as “Weak Gross Substitutability” (WGS) [21], but the problem can be PPAD-hard when demand responses display any “non-monotonicity” (a property slightly stronger than non-WGS) [20]. Simple and natural utility functions can fall on either side of this complexity gap; therefore, a real-world deployment of a batch trading system must ensure that (potentially adversarial) user-submitted CFMM trading functions satisfy WGS. We find a natural class of utility functions that admit a natural description format that is easy to validate and sufficient but not necessary to guarantee WGS.

Formulation of market equilibrium as the solution of a convex program has led to improved results in computational efficiency and structural understanding of markets in other contexts (such as the much-celebrated result of Eisenberg and Gale [29]). Therefore, we give a convex program (§6) for computing the equilibria in batch exchanges which include CFMMs. This program may be of independent interest – an equivalent statement is that this program computes equilibria in Arrow-Debreu exchange markets where agent utility functions are arbitrary quasi-concave functions of two assets, subject to the constraint that demand responses satisfy WGS.

Our convex program is inspired by that of Devanur et al. [24] for linear utility functions. The proof is technical, but the intuition is easy to state – we develop a viewpoint from which arbitrary utility functions appear as an uncountable collection of infinitesimal agents with linear utility functions.

When the density of this infinite collection of limit offers is a rational linear function, we prove that our convex program for market equilibria has rational solutions. However, some CFMMs (such as those based on the Logarithmic Market Scoring Rule [37]) can force batches, even with only two agents, to only admit irrational solutions.

1.2 Related Work

Automated market-making strategies have been studied both in a blockchain context [8] and in traditional exchanges [34, 35, 41, 46]. CFMMs form a subclass of automated market-making. There has been extensive study on the design of CFMM trading functions [8, 10, 28] and how the design of a CFMM trading function interacts with the economic incentives of those who invest in it [17, 18, 30, 31, 36, 45].

The economics behind batch trading schemes are studied in [11, 15, 16, 42]. The well-studied model of Arrow and Debreu [13] forms the basis for batch trading implementations [40, 48]. There are many classes of algorithms for (approximately or exactly) computing equilibria in Arrow-Debreu exchange markets, including iterative methods [14, 21, 22, 32], convex programs (for the case of linear utilities) [24], primal-dual path following [49], and combinatorial algorithms [26, 27, 33, 38, 39]. Devanur and Vazirani [25] extend the exchange market model to so-called “spending constraint” utilities, where an agent measures utility not from the amount of a good purchased, but from the fraction of its budget spent on the good.

2 PRELIMINARIES

2.1 Batch Exchanges

In a batch exchange, an exchange operator gathers together requests to trade, and then satisfies some (or all) of those requests at once. Mechanisms for batch trading vary — one might, for example, compute a uniform clearing price [16], possibly subject to certain regulatory constraints [15], or solve a custom optimization problem [2]. These mechanisms all share, however, a common underlying framework.
In this work, all of the assets discussed will be divisible and fungible. As a convention, for a quantity or trade \(\{x_A\}_{A \in \mathcal{A}}\), we typically write \(x \geq 0\) to mean \(x_A \geq 0\) for all \(A \in \mathcal{A}\).

**Definition 2.1.** A Batch Exchange that facilitates trades between a set of assets \(\mathcal{A}\) consists of an exchange operator and a set of participants, each with an endowment of goods \(e = \{e_A\}_{A \in \mathcal{A}}\) (with \(e_A \geq 0\)) and a “utility” function \(u : \mathbb{R}^{|\mathcal{A}|} \rightarrow \mathbb{R}\). We say that a set of bundles of assets \(x_A \in \mathbb{R}^{|\mathcal{A}|}\) settles a batch if, for each \(i\), \(u_i(x_i) \geq u_i(e_i)\) (the participant accepts the trade) and \((x_1)_A \geq 0\) (the participant’s balance of each asset is nonnegative).

In effect, each user receives a trade of \(x_1 - e_1\).

A real-world deployment of a batch exchange would consist not of one single batch settlement, but rather, many batch settlements executed sequentially over time. We make no assumptions about the relationship between one batch settlement and the next. The discussion here focuses on one batch, in isolation.

This work is primarily interested in batch exchanges which are modeled as instances of Arrow-Debreu exchange markets; Definition 2.1 and the axioms of §3, however, are written in more general language to accommodate objects like CFMMs that a priori do not obviously correspond to an agent with a utility function. For example, the “utility function” of Definition 2.1 could be the indicator function of a set of Boolean constraints—CFMMs in practice are, for example, typically implemented as smart contracts that check a Boolean predicate on a proposed trade. The goal of §4 is to understand what possible utility functions capture the behavior of a CFMM (as specified in Axiom 5), while also satisfying the axioms of §3.1. Note that there may not always be a set of trades that settles a batch subject to these axioms.

**Example 2.2.** A traditional limit sell offer, which sells 1 unit of \(A\) in exchange for \(B\) for a minimum price of \(k\) \(B\) per \(A\), maps to a participant with an endowment of 1 unit of \(A\) and a utility function \(u(x_A, x_B)\) which is at least \(u(1, 0)\) if \(\frac{x_B}{x_A} \geq k\) and \(x_B \geq 0\), and strictly less than \(u(1, 0)\) otherwise.

### 2.2 Arrow-Debreu Exchange Markets

An Arrow-Debreu exchange market [13] is a game between a set of traders and a marketplace. Each trader \(i\) has an endowment of goods \(x_i \in \mathbb{R}_{\geq 0}\) and a utility function \(u_i : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}\).

An equilibrium of the market is a set of prices \(\{p_A\}_{A \in \mathcal{A}}\) and, for each agent \(i\), a collection of goods \(x_i\) such that \(x_i\) maximizes \(u_i(x)\) subject to a “budget constraint” \(p \cdot x_i \leq p \cdot e_i\). At equilibrium, for each asset \(A\), \(\Sigma_i (e_i - x_i)_A \geq 0\), and either the inequality is tight or \(p_A = 0\).

Conceptually, each agent independently sells their endowment to the “market” in exchange for a virtual “money”, and buys back its preferred, optimal bundle of goods.

An Arrow-Debreu exchange market does not always admit an equilibrium, and an equilibrium may not always have a positive price for each asset (similarly, Definition 2.1 does not guarantee that a batch can be settled). This is a property of agent utility functions (resp. batch predicates), but under mild conditions (i.e. condition + of [24]), an equilibrium with positive prices always exists. All of the instances studied in this work (in §6) admit equilibria with positive prices.

As an example, the behavior of a limit sell offer can be captured as an agent in an Arrow-Debreu exchange market maximizing a linear utility function.

**Example 2.3.** An Arrow-Debreu exchange market agent with an endowment of 1 unit of \(A\) and a utility function of \(u(x_A, x_B) = x_A + k x_B\) trades \(A\) for \(B\) if \(\frac{p_A}{p_B} > k\) and does not if \(\frac{p_A}{p_B} < k\).

1Note that this linear function satisfies the requirements for the utility function in Example 2.2.
2.3 Constant Function Market Makers

A Constant Function Market Maker (CFMM) is an automated market-making strategy parameterized by a trading function \( f(\cdot) \). At any time, the CFMM owns some assets (its “reserves”). We denote CFMM reserves by \( x = \{ x_A \}_{A \in \mathbb{A}} \), for some set of assets \( \mathbb{A} \) and each \( x_A \in \mathbb{R}_{\geq 0} \). Reserves are provided by deposits from investors participating in liquidity provision (so-called “liquidity providers”).

The trading function determines whether a CFMM accepts a proposed trade. A CFMM with reserves \( x \) and function \( f(\cdot) \) accepts any trade that results in reserves \( x' \) (i.e. a trade of \( x' - x \)), so long as \( f(x') \geq f(x) \). In practice, rational traders trading with the CFMM only make trades that leave \( f(x') = f(x) \).

In this work, we make the standard assumption (e.g. [8]) that all trading functions are continuous, nondecreasing, and quasi-concave on the positive orthant. The (sub)gradient of the trading function gives the exchange rate that a CFMM offers for a trade of infinitesimal size.

Definition 2.4 (Spot Valuations). A set of asset valuations \( \{ p_A \}_{A \in \mathbb{A}} \) is a set of spot valuations for a CFMM with trading function \( f(\cdot) \) at reserves \( \hat{x} \) if \( p \) is a subgradient of \( f(\cdot) \) at \( \hat{x} \).

When a trading function \( f \) is differentiable (and partial derivatives are nonzero), then there is a unique spot exchange rate from \( \mathcal{A} \) to \( \mathcal{B} \) of \( \frac{\partial f_A}{\partial f_B} \).

3 AN AXIOMATIC APPROACH TO CFMMs IN BATCH TRADING SCHEMES

We give here a set of natural axioms that capture a minimal set of desirable properties in batch exchanges. All of these axioms are satisfied in exchanges modeled as Arrow-Debreu exchange markets, but these are not the only ways to run a batch exchange. We briefly discuss after each axiom what properties the axiom gives and what properties differ without the axiom.

Note that these axioms here implicitly categorize also the utility functions used in a batch exchange, in the sense that it is possible for certain utility functions to prevent the existence of a set of trades that settles the batch subject to these axioms. The axioms below are written to be conditioned on the existence of a batch settlement. Corollary A.3 relates existence of a batch settlement to existence of an equilibrium of an Arrow-Debreu exchange market with positive prices, which Lemma A.4 shows exists given Assumptions 1 and 3.

3.1 Axioms of Batch Trading

3.1.1 Uniform Valuations

Axiom 1 (Uniform Valuations): Let \( \{ x_i \} \) be a set of trades that settle a batch exchange. There exists a set of shared asset valuations \( \{ p_A \geq 0 \}_{A \in \mathbb{A}} \), such that \( (x_i - c_i) \cdot p \leq 0 \) for all participants \( i \).

Implicitly, a set of asset valuations implies an exchange rates between every asset pair That is to say, the exchange rate from \( \mathcal{A} \) to \( \mathcal{B} \) is \( \frac{p_A}{p_B} \). These rates are free of cyclic arbitrage; that is, \( \frac{p_A}{p_B} = \frac{p_A}{p_C} \frac{p_C}{p_B} \). We write Axiom 1 as an inequality for full generality, (allowing a participant to trade at or below the implied exchange rates), but Proposition 4.2 combines this axiom with Axiom 2 to show that the inequality must be tight.

This axiom is the core of what makes a batch exchange potentially different from a traditional market. Fairness properties largely follow from this axiom—if every participant sees the same exchange rates, the exchange operator
cannot directly privilege one participant over another. Similarly, uniform exchange rates eliminate latency arbitrage and the premium for high-frequency trading present in traditional, continuous double-auctions [16].

3.1.2 Asset Conservation.

**Axiom 2** (Asset Conservation): Let \( \{x_i\} \) be a set of trades that settle a batch exchange. Then for all assets \( A \), \( \Sigma_i (x_i)_A = 0 \).

An exchange, generally, should not create or destroy units of an asset. This does not preclude an exchange operator from also acting as a participant in the batch exchange; for example, Kyle [41] studies a model of a batch exchange where one (deep-pocketed) market-maker participates in the batch and sets the batch prices.

3.1.3 Local Optimality. One beneficial property of the Arrow-Debreu exchange market model is that every agent independently maximizes its own utility function. This ensures that every agent always gets what it considers an optimal deal, conditioned on the market prices. An agent’s optimal deal may not be unique; in this case, a market equilibrium specifies one of the set of optimal trades for the agent. But whether or not a deal is optimal for agent depends only on the agent and the market prices, not on the choices of any other agent.

Analogously, in a batch exchange, every participant should independently get what it considers its “best” deal, given the uniform batch valuations. Definition 2.1 does not assume any particular structure to participant utility functions (they could be, for example, the indicator function of a set of Boolean constraints), but if the function does imply a preference between two potential trades, the batch participant should get their preferred trade.

**Axiom 3** (Local Optimality): Let \( \{x_i\} \) be a set of trades that settles a batch exchange. For each participant \( i \), the trade \( x_i \) should maximize \( u_i(\cdot) \), subject to the constraint of Axiom 1 (that is, \( (x_i - e_i) \cdot p \leq 0 \)).

This axiom extends the part of Definition 2.1 that states that a batch participant never receives a trade that strictly reduces the value of its utility function.

3.1.4 Pareto Optimality. After a batch exchange settles a batch of trades, there should not be a Pareto-improving further set of trades. One beneficial property about batch exchanges modeled as Arrow-Debreu exchange markets is that subject to mild conditions on agent preferences (Lemma A.1), the fact that every agent receives a trade that they individually consider optimal suffices to guarantee that there are no further Pareto-improving trades.

Semantically, participants in a batch exchange trade with the batch, not with each other. As such, participants should be agnostic to the preferences of other participants, and should be guaranteed that after a batch, they need not pursue further deals between coalitions of other agents.

The following axiom asserts this property.

**Axiom 4** (Pareto Optimality): Let \( \{x_i\} \) be any set of trades that settles a batch exchange subject to Axioms 1 and 3. There is not any set of additional trades \( \{y_i\} \) such that \( \Sigma_i y_i = 0, x_i + y_i \geq 0 \) and \( u_i(x_i + y_i) \geq u_i(x_i) \) for each \( i \), and \( u_i(x_i + y_i) > u_i(x_i) \) for at least one \( i \).

This axiom precludes batch exchange designs where, for example, whether or not a limit sell order trades depends on the presence or absence of a CFMM trading the same assets.

3.2 Axioms of CFMMs

We axiomatize CFMMs according only to the constraints that these objects enforce in practice. Per its name, a CFMM should accept any trade if the trade keeps constant the value of its trading function.
One might consider encoding this strict equality condition as an axiom; however, real-world CFMM deployments only check the weaker condition that the trading function’s value does not decrease [3], and assume that rational users will submit trades to the CFMM that keep the inequality tight. In fact, we illustrate in Example 3.1 that a strict equality condition is too strong to allow CFMMs to integrate meaningfully in a batch exchange (while satisfying the axioms of §3.1).

**Axiom 5 (Non-Decreasing Trading Functions):** A CFMM $i$ with reserves $e_i$ and trading function $f_i$ accepts a trade $x_i$ if and only if $f(x_i) \geq f(e_i)$.

When modeled as a batch participant with some utility function $u_i(\cdot)$ this axiom means that $u_i(\cdot)$ must satisfy the property that if $x_i \geq 0$ and $f(x_i) \geq f(e_i)$, then $u_i(x_i) \geq u_i(e_i)$. Naturally, the choice $u_i(\cdot) = f(\cdot)$ satisfies this condition, but so too does the indicator function mapping some set of assets $y$ to 1 if and only if $f(y) \geq f(e_i)$.

Example 3.1 illustrates how the stricter axiomatization of CFMM behavior (i.e. $f(x_i) = f(e_i)$) can easily force a CFMM to not make any trades at all, thereby undermining its function as a market-maker (in general, this phenomenon follows from the quasi-concavity of $f(\cdot)$ and Proposition 4.3).

**Example 3.1.** Consider the following batch trading instance with one CFMM and two limit sell orders (Example 2.3).

- The first limit order sells up to 1 unit of $A$ for $B$ with a minimum price of 1 $B$ per $A$.
- The second sells up to 3 units of $B$ in exchange for $A$, with a minimum price of 1/6 $A$ per $B$.
- The CFMM is a constant-product market maker ($f(A, B) = A \cdot B$) with reserves of 1 unit of $A$ and 10 units of $B$.

If this instance follows Axioms 1, 2, and 3, and preserves $\mathcal{F}$ and only check the weaker condition that the trading function’s value does not decrease [3], and assume that rational users...
4.1 Non-Satiation

Ensuring Pareto-optimality requires, at minimum, that batch participants be unwilling to freely give away valuable assets.

**Proposition 4.2.** Axioms 1 and 2 imply that if a set of trades \( \{x_i\} \) settles an instance of a batch exchange at valuations \( \{p,A\} \), then \( p \cdot x_i = p \cdot e_i \) for all \( i \).

*Proof.* By Axiom 2, \( \Sigma_i (p \cdot x_i) = 0 \). If there is some \( i \) with \( p \cdot x_i < p \cdot e_i \), then there must be some other \( i' \) with \( p \cdot x_{i'} > p \cdot e_{i'} \), which violates Axiom 1.

As such, Proposition 4.2 rules out Rule A above. Furthermore, any utility function \( u(\cdot) \) used within a batch exchange must only consider trades exactly at the batch prices to be optimal.

**Proposition 4.3.** Axioms 1, 3, and 4 imply that for any batch valuations \( p \) and any utility function \( u_i(\cdot) \) used in a batch exchange with some endowment \( e_i \), the set of trades \( X_i = \{x_i\} \) that maximize \( u_i(x_i) \) subject to \( x_i \geq 0 \) and \( p \cdot x_i \leq p \cdot e_i \) must all have \( p \cdot x_i = p \cdot e_i \) if there exists some \( x_i' \in X_i \) with \( x_i' \cdot p = p \cdot e_i \).

*Proof.* Suppose not. Consider a utility function \( u_i(\cdot) \), endowment \( e_i \), and prices \( p \) such that there exists \( x_i \) maximizing \( u_i(x_i) \) subject to \( x_i \geq 0 \) and \( p \cdot x_i \leq p \cdot e_i \). Furthermore, suppose that in fact, \( p \cdot x_i < p \cdot e_i \), and that there exists \( x_i' \) maximizing \( u_i(x_i') \) subject to the same conditions with \( p \cdot x_i' = 0 \).

Consider a batch exchange instance with one agent with endowment \( e_i \) and utility \( u_i(\cdot) \) as above, and one agent with linear utility function \( u_j(y) = p \cdot y \) and endowment \( e_j = \{\max((e_i, A), 0)\}_{A \in \mathcal{W}} \). Then the bundles of goods \( x_i' \) for the first agent and \( e_j - (x_i' - e_i) \) for the second and batch valuations \( p \) settles the batch subject to the conditions of Axioms 1 and 3 (and Axiom 2). However, such a set of trades admits a Pareto improvement, violating Axiom 4 (in which the first participant receives a further trade of \( x_i - x_i' \), and the second receives an additional \( x_i' - x_i \)).

4.2 Align Batch Valuations and CFMM Spot Valuations

We now turn to the case of the utility function \( u(\cdot) \) of a CFMM with trading function \( f_i(\cdot) \) in a batch exchange. In particular, the question at hand is to characterize, for a given set of valuations \( p \) and the CFMM’s endowment \( e_i \), which trades \( x_i \) can maximize \( u_i(\cdot) \) without creating the possibility of a violation of Axiom 4.

Axiom 4 is, in effect, an axiom about the types of utility functions that a batch exchange system can accept as participants. A batch exchange deployment satisfying Axiom 4, therefore, would have to restrict the types of participants, and certainly, one can construct batch exchange instances using pathological utility functions and a CFMM that never admit a Pareto-optimal set of trades that settles the batch, no matter the utility function used to model the CFMM (i.e. by a Pareto-improving set of trades that does not involve the CFMM).

At minimum, however, a batch exchange should handle limit sell offers (i.e. Example 2.2), corresponding to participants with linear utility functions. And in fact, by Lemma A.1, a batch exchange with limit sell orders that satisfies Axioms 1, 2, and 3 always satisfies Axiom 4.

We therefore focus here on the minimal case of a batch exchange incorporating linear utility functions and one CFMM. Lemma 4.4 characterizes which trades a CFMM (satisfying Axiom 5) can make in such a batch exchange without violating Axiom 4. These trades are exactly those that maximize a CFMM’s trading function. Lemma A.1 then implies directly that any batch exchange that satisfies Axioms 1, 2, and 3 where participants have either linear utilities or are CFMMs using this trading mechanism must also satisfy Axiom 4.
**Lemma 4.4.** Let \( u_i() \) be the utility function of a CFMM with endowment \( e_i \) and trading function \( f_i() \). Then, for any set of valuations \( p \), if there exists a trade \( x_i \) such that \( p \cdot x_i = p \cdot e_i, x_i \geq 0, \) and \( x_i \) maximizes \( u_i(x_i) \) but not \( f_i(x_i) \) (subject to \( x_i \geq 0 \) and \( p \cdot x_i \leq p \cdot e_i \), then there exists a batch exchange instance (that satisfies Axioms 1, 2, 3, and 5) involving the CFMM and one participant with a linear utility function that violates Axiom 4.

**Proof.** Suppose there exists a set of valuations \( p \) and a trade \( \hat{x}_i \) with \( \hat{x}_i \cdot p = p \cdot e_i \) that maximizes \( u_i() \) but not \( f_i() \) subject to \( \hat{x}_i \geq 0 \) and \( \hat{x}_i \cdot p \leq p \cdot e_i \).

Let \( Y = \{ y \} \) be the set of additional trades such that \( f_i(\hat{x}_i + y) \geq f_i(\hat{x}_i), (\hat{x}_i + y) \cdot p \leq 0, \) and \( \hat{x}_i + y \geq 0 \). \( Y \) must be convex (by the quasi-concavity of \( f_i() \)), closed (by continuity of \( f_i() \)) and nonempty. Furthermore, there must be some \( \hat{y} \in Y \) with \( p \cdot \hat{y} \neq 0 \). For example, if \( z \) maximizes \( f_i() \), then there exists a real \( 0 < \alpha \leq 1 \) such that \( f_i((1 - \alpha)z) = f_i(\hat{x}_i) \) (so \( -\alpha z \in Y \)).

Consider a batch exchange instance containing this CFMM and one participant with a linear utility function of \( u_j(x_j) = p \cdot (x_j) \) and endowment of \( |(e_j)_{\mathcal{A}}| = |(\hat{x}_j)_{\mathcal{A}}| + |(\hat{y})_{\mathcal{A}}| \) \( \forall \mathcal{A} \in \mathbb{Q} \).

Then the bundle of goods \( \hat{x}_i \) for the CFMM and \( e_j - (\hat{x}_i - e_i) \) for the other participant is a settlement for the batch that satisfies Axioms 1, 2, and 3. However, there exists a set of trades (\( \hat{y} \) for the CFMM, \( -\hat{y} \) for the other participant) that Pareto-improves the settlement, violating Axiom 4 (Axiom 5 implies that the CFMM’s utility function must not decrease when performing the additional trade of \( \hat{y} \)).

By contrast, using the trading function as a CFMM’s utility function ensures that a batch exchange can satisfy Axiom 4.

**Theorem 4.5.** Any batch exchange system where participants are either limit sell orders (using linear utility functions) or CFMMs (using the trading function as the utility function) satisfies Axiom 4 (Pareto Optimality). If a settlement satisfies Axiom 3 (Local Optimality), then the CFMM’s behavior satisfies Axiom 5 (Non-Decreasing Trading Functions).

**Proof.** When a CFMM’s trading function is used as its utility function in a batch, maximizing that utility function cannot cause the utility function’s value to decrease (when the starting reserves \( e_i \) are a feasible input). Axiom 4 follows from Lemma A.1. □

## 5 CFMM Trading Function Structure and Computational Efficiency

Implementing a batch exchange subject to the axioms of §3 requires computation of equilibria of Arrow-Debreu exchange markets (Corollary A.3). However, the computational complexity of computing equilibria in these markets strongly depends on the agent’s utility functions [22]; some natural classes of utility functions make the problem of computing an equilibrium PPAD-complete [19]. Per Theorem 4.5, then, the complexity of computing batch equilibria depends on the CFMM trading functions.

Utility functions of limit sell offers satisfy a property known as Weak Gross Substitutability (WGS) [12] (Mirroring the discussion in [22], the definition below refers only to strictly quasi-concave utility functions, for which an agent’s best response is unique).

**Definition 5.1.** Consider two sets of prices \( p_1 \) and \( p_2 \) with \( p_1 \leq p_2 \), and any sets of assets \( x_1 \geq 0 \) and \( x_2 \geq 0 \), where \( x_i \) is the best response of an agent with strictly quasi-concave utility function \( u() \) and endowment \( e \) at prices \( p_i \).

The utility function satisfies WGS if, for any asset \( \mathcal{A} \), if \( (p_1)_{\mathcal{A}} = (p_2)_{\mathcal{A}} \), then \( (x_1)_{\mathcal{A}} \leq (x_2)_{\mathcal{A}} \).

In other words, decreasing the valuations of some assets does not decrease demand for assets whose prices are fixed.
WGS holds on a set of agents if an analogous property holds on the aggregation of a set of best responses, and naturally holds on a set if it holds on every agent’s utility function. It follows from Theorem 7 of Codenotti et al. [21] that equilibria can be approximated in polynomial time when a market satisfies WGS, but by Theorem 7 of Chen et al. [20], the problem of approximately computing equilibria in exchange markets where (some groups of) agents display a (stronger) “non-monotonicity” property is PPAD-complete.

Even the seemingly natural limit buy offer (Example 5.2), which buys a fixed amount of some asset for as low a price as possible (if the price is sufficiently low), can be used to encode the non-monotonic behavior that causes PPAD-completeness.

Example 5.2. An Arrow-Debreu exchange market agent with an endowment of 1 unit of $A$ and a utility function of $u(x_A, x_B) = x_A + k \min(1, x_B)$ trades its $A$ to buy at most 1 unit of $B$ if $\frac{p_A}{p_B} > k$ (selling as little $A$ as possible) and does not if $\frac{p_A}{p_B} < k$.

Proposition 5.3. The problem of computing equilibria in batches of limit sell offers is solvable in polynomial time, but the problem of computing equilibria in batches that include limit buy and limit sell offers is PPAD-complete.

More generally, exchange market equilibria can be approximated in polynomial time if the utility function of every market participant satisfies WGS. However, the problem of computing equilibria in markets that contain non-monotone utility functions is PPAD-complete.

5.1 Efficient CFMM Trading Function Validation

Therefore, a batch exchange implementation needs an efficient way to validate that any user-supplied CFMM trading function satisfies WGS, and this validation should hold for any initial endowment of assets (i.e. a CFMM should not need to be checked repeatedly as its assets change over a sequence of batches). Otherwise, adversarial users could potentially make the equilibrium computation problem intractable and impair the operation of a batch exchange implementation.

The best response of a CFMM at any set of batch valuations should also be easily computable.

We outline below a flexible, structured format for expressing utility functions that ensures that functions always satisfy WGS, regardless of the associated endowment of goods.

Definition 5.4 (Spending Specification). A subclass of utility functions can be specified by the fraction of their budget $s_A(p) \rightarrow [0, 1]$ that they spend on each good $A \in \mathcal{A}$. A specification must have $\Sigma_A s_A(p) = 1$ for all valuations $p$, and each $s_A(\cdot)$ must be continuous.

Specifically, the optimal response of an agent with endowment $e$ at prices $p$ is to buy $\frac{e \cdot p}{p_A} s_A(p) = 1$ units of $A$.

Not all utility functions can be specified in this manner.3 Specifically, such a specification requires that for all sets of goods $x \geq 0$, if $x$ maximizes a utility function $u(\cdot)$ subject to $x \cdot p \leq k$ for some valuations $p$ and budget $k$, then $ax$ maximizes $u(\cdot)$ subject to $(ax) \cdot p \leq (ak)$ for all $a > 0$.

However, for those that can, it suffices for a batch exchange to check once the following condition.

Lemma 5.5. Suppose a set of functions $\{s_A(\cdot)\}$ specify the optimal spending of some utility function (as in Definition 5.4).

---

2The approximation is to a $(1 + \varepsilon)$ level of accuracy for any $\varepsilon > 0$ and the natural metric of accuracy given in Definition 1 of [21]. "Polynomial time" means polynomial in the size of the input market instance and $1/\varepsilon$.

3Nor does a specification in this manner guarantee that every utility function that could match the specification is quasi-concave. However, as discussed in A.1, standard arguments based on Kakutani’s fixed point theorem showing the existence of equilibria in Arrow-Debreu exchange markets can be naturally modified to admit agents that behave according to a spending specification.
If, for every \( s_A(\cdot) \) and for any \( p' < p \) with \( p_A = p'_A \), \( s_A(p) \geq s_A(p') \), then the behavior of an agent using this utility function satisfies WGS.

**Proof.** Let \( p \) and \( p' \) be as in the lemma, and let \( e \) be any endowment. The demand of the agent for \( A \) at \( p \) is \( \frac{p}{p_A} s_A(p) \), which is at least \( \frac{p}{p_A} s_A(p') \).

\[ \frac{p}{p_A} s_A(p) \geq \frac{p}{p_A} s_A(p'), \quad \square \]

This check is feasible for CFMMs that trade 2 assets (say \( A \) and \( B \)), as it suffices to specify the spending fraction for only one asset with some function of one real value \( (\frac{p_A}{p_B}) \) that can be statically shown to be increasing (i.e. a piecewise linear function with positive derivatives, or a polynomial with nonnegative coefficients). Furthermore, if this function is efficiently computable, then the CFMM’s demand response in a batch exchange is efficiently computable (as opposed to e.g. directly maximizing a utility function via convex optimization). Some algorithms \([14, 21]\) rely on repeated queries to agent demand responses.

Some widely used CFMM trading functions can be expressed in this manner.

**Example 5.6.**

1. The constant product trading function \( u(x_A, x_B) = x_A x_B \) spends half of its budget on each good.
2. The weighted constant product trading function \( u(x_A, x_B) = x_A w_a x_B w_b \) for positive constants \( w_a, w_b \) spends a \( \frac{w_a}{w_a + w_b} \) fraction of its budget on \( A \).

6 A CONVEX PROGRAM FOR 2-ASSET WGS UTILITY FUNCTIONS

Our contribution here is a convex program that computes equilibria in batch exchanges that incorporate CFMMs that trade between two assets and that satisfy WGS. Equivalently, this program computes equilibria in Arrow-Debreu exchange markets where every agent’s behavior satisfies WGS and every agent has utility for only two assets. The program is based off of the convex program of Devanur et al. \([24]\) for linear exchange markets — that is, when a batch exchange contains only limit sell offers.

The key observation is that 2-asset CFMMs satisfying WGS can be viewed as (uncountable) collections of agents with linear utilities and infinitesimal endowments. This correspondence lets us replace a summation over agents with an integral over this collection of agents. However, proving correctness of our program requires direct application of Kakutani’s fixed point theorem, instead of the argument based on Lagrange multipliers used in \([24]\).

Smooth and strictly quasi-concave CFMMs lead to smooth objective functions. Gradients are easy to compute for many natural CFMMs. The runtime per gradient computation is linear in the number of assets and in the number of CFMMs.

6.1 From a Utility Function to a Continuum of Limit Orders

Suppose that an exchange market agent is ever only interested in two assets \( A \) and \( B \). We first give a viewpoint that corresponds such an agent to a continuum of agents with linear utility functions that trade between \( A \) and \( B \). Such an agent’s optimal behavior can only depend on the exchange rate between the two assets (that is, \( r_A B = \frac{p_A}{p_B} \)). We can therefore define a function \( D_A(\cdot) \) that gives, for any exchange rate \( r \), the amount of \( A \) that the agent sells (relative to its initial endowment).

**Definition 6.1 (Cumulative Density).** Consider some agent with endowment \( e \) and utility function \( u(\cdot) \) that only ever has marginal utility for assets \( A \) and \( B \).
As such, $d_{12}$

Geoffrey Ramseyer, Mohak Goyal, Ashish Goel, and David Mazières

$D$ density function $D$ regarding the above decomposition).

behaves in the same manner as a continuum of these marginal limit sell offers (and perhaps some finite-sized offers,

$B$ for $B$

give jump discontinuities of the above form.

such a manner. Note that strict quasi-concavity implies a point-valued density function, and linear utility functions

another (not just the two

agent's net trading behavior is expressible as a sum of arbitrarily many density functions trading from one asset to

Continuity implies that each $D_i(\cdot)$ has a closed graph, and quasi-concavity (implied by part 3) implies that each $D_i(r)$

is a closed set.

Definition 6.2 (Inverse Cumulative Density). $D_i^{-1}(x)$ is the least upper bound on the set $\{ r | D_i(r) \leq x \}$. When the

set is empty, $D_i^{-1}(x) = 0$, and $\infty$ when the set is unbounded.

We make the following simplifying assumption in the rest of the discussion.

Assumption 2. Every density function $D_i(\cdot)$ is either a monotonic, continuous, point-valued function or a threshold

function; for some constants $r_i$ and $d_i$, $D_i(r) = 0$ for all $r < r_i$, $D_i(r) = d_i$ for $r > r_i$, and $D_i(r_i) = [0,d_i]$. Furthermore, $D_i(\infty) > 0$.

This is without loss of generality when density functions are monotonic. The results below only requires that an

agent’s net trading behavior is expressible as a sum of arbitrarily many density functions trading from one asset to

another (not just the two $D_{A_i}(\cdot)$ and $D_{B_i}(\cdot)$). Monotonicity means that every density function can be decomposed in

such a manner. Note that strict quasi-concavity implies a point-valued density function, and linear utility functions

give jump discontinuities of the above form.

Definition 6.3 (Marginal Density function). The marginal density function $d(r)$ of an agent selling $A$ in exchange for $B$ is

$\frac{\partial D_i(r)}{\partial r}$. Intuitively, the marginal density function represents the marginal amount of $A$ that an agent sells at a given price. As such, $d(r)$ is conceptually the size of a limit sell offer with minimum price $r$, and an agent with density function $D(r)$ behaves in the same manner as a continuum of these marginal limit sell offers (and perhaps some finite-sized offers, regarding the above decomposition).

Monotonicity holds when an agent’s behavior satisfies Weak Gross Substitutability.

Lemma 6.4. An agent trading between 2 assets whose utility function is strictly quasi-concave has monotonic cumulative density function $D(r)$ for each asset if and only if its optimal behavior satisfies WGS.

The Cumulative (Sell) Density of the agent for asset $A$ at exchange rate $r > 0$ is $D_A(r) = \{ \max(0, e_A - x_A) \}$ for $x$ any optimal response vector for the agent at exchange rate $r$.

A density function is monotonic if, for any $r_1 < r_2$ and $x_1, x_2 \in D(r_1), x_2 \in D(r_2)$, then $x_1 \leq x_2$.

We require that $D_i(\cdot) < \infty$ (asset amounts are finite).

The budget constraint is zero-sum between $A$ and $B$, so for every exchange rate $r$, either $D_A(r) = 0$ or $D_B(r) = 0$. In the

rest of this section, it will be convenient to consider each density function separately. Specifically, we will assume

that an exchange market consists of a set of density functions, for each each function $i$ sells good $A_i$ and buys $B_i$ (and therefore write only $D_i(\cdot)$, when clear from context).

We also make the following assumption on utility functions (reproduced exactly from the assumptions in Arrow and Debreu’s original work [13]).

Assumption 1. A utility function $u(\cdot)$ satisfies the following properties.

1. $u(\cdot)$ is continuous.
2. For all endowments $e$, there exists $e'$ with $u(e) < u(e')$.
3. For any $e, e'$ with $u(e) < u(e')$ and any $0 < t < 1, u(e) < u(te + (1-t)e')$.

This is without loss of generality when density functions are monotonic. The results below only requires that an

agent’s net trading behavior is expressible as a sum of arbitrarily many density functions trading from one asset to

another (not just the two $D_{A_i}(\cdot)$ and $D_{B_i}(\cdot)$). Monotonicity means that every density function can be decomposed in

such a manner. Note that strict quasi-concavity implies a point-valued density function, and linear utility functions

give jump discontinuities of the above form.

Definition 6.3 (Marginal Density function). The marginal density function $d(r)$ of an agent selling $A$ in exchange for $B$ is

$\frac{\partial D_i(r)}{\partial r}$. Intuitively, the marginal density function represents the marginal amount of $A$ that an agent sells at a given price. As such, $d(r)$ is conceptually the size of a limit sell offer with minimum price $r$, and an agent with density function $D(r)$ behaves in the same manner as a continuum of these marginal limit sell offers (and perhaps some finite-sized offers, regarding the above decomposition).

Monotonicity holds when an agent’s behavior satisfies Weak Gross Substitutability.

Lemma 6.4. An agent trading between 2 assets whose utility function is strictly quasi-concave has monotonic cumulative density function $D(r)$ for each asset if and only if its optimal behavior satisfies WGS.
When possible, we use the same notational conventions as in [24]. We assume that a set of agents has utility functions that imply a set of density functions that satisfy Assumption 2, with \( D_i() \) for \( i \in N \) continuous and \( D_i() \) for \( i \in M \) threshold functions. These agents trade a set of assets \( A \).

Variables \( p_A \) denote the valuation of good \( A \), and the variable \( y_i \) for \( i \in N \cup M \). At equilibrium, the quantity \( x_i \) denotes the amount of good \( A \) that density function \( i \) sells to the market, receiving \( \frac{p_A}{p_B} x_i \) units of \( B_i \) (so at an exchange market equilibrium, \( x_i = D_i(\frac{p_A}{p_B}) \)). Note that quantities \( x_i \) do not appear in the program; instead, we use the variables \( y_i \) that denote the “value” traded by agent \( i \); that is, \( y_i = p_A x_i \). Define \( \beta_i(z) = \min(p_A i, p_B i, z) \). Informally, \( \beta_i(z) \) is the inverse best bang-per-buck for the marginal limit sell order of agent \( i \) at exchange rate \( z \). Finally, for continuous density functions, define \( g_i(x_i) \) to be \( \int_0^{D_i^{-1}(x_i)} d(r) \ln(1/r) \, dr \). For threshold density functions, define \( g_i(x_i) = \min(d_i, x_i) \ln(1/r_i) \).

**Lemma 6.5.** \( g_i() \) is a concave function, and \(-p_A g_i(y_i/p_A)\) is convex.

**Proof.** For continuous density functions,

\[
\frac{\partial}{\partial x} g_i(x) = \frac{\partial}{\partial x} \int_0^{D_i^{-1}(x)} d(r) \ln(1/r) \, dr = d(D_i^{-1}(x)) \ln \left( \frac{1}{D_i^{-1}(x)} \right) \frac{1}{d(D_i^{-1}(x))} = \ln \left( \frac{1}{D_i^{-1}(x)} \right)
\]

Because the derivative of \( g_i \) is a decreasing function, \( g_i \) must be concave (for \( x \geq 0 \)). Concavity clearly holds for threshold density functions.

\(-p_A g_i(y_i/p_A)\) is the perspective transformation of a convex function, so it is convex.

**Theorem 6.6.** The following program is convex and always feasible. Its objective value is always non-negative. When the objective value is 0, the solution forms an exchange market equilibrium with nonzero prices, and when such an equilibrium exists, the objective value is 0.

\(^{4}\beta_i() \) corresponds to the \( \beta \) variables of [24].

\(^{5}\)The expression \(-p_A g_i(y_i/p_A)\) recovers an analogue of the \(-y_i \ln(u_i)\) term in [24].
Minimize \( \sum_{i \in N} P_{A_i} \int_0^\infty \left( d_i(z) \ln \left( \frac{P_{A_i}}{\hat{\beta}_{i,z}(p)} \right) \right) dz + \sum_{i \in M} P_{A_i} d_i \ln \left( \frac{P_{A_i}}{\hat{\beta}_{i,r_i}(p)} \right) - \sum_{i \in N \cup M} P_{A_i} g_i(y_i/p_{A_i}) \)

Subject to \( \sum_{i : A_i = C} y_i = \sum_{i : B_i = C} y_i \) \( \forall C \in \mathcal{C} \)

\( P_C \geq 1 \) \( \forall C \in \mathcal{C} \)

\( 0 \leq y_i \leq P_{A_i} D_i(\infty) \) \( \forall i \in N \cup M \).

**Proof.** Lemma 6.7 shows the convexity and feasibility of the program. Lemma 6.8 shows that the objective is nonnegative. Lemma 6.9 shows that the objective value is 0 if and only if the optimal solutions satisfy \( y_i = P_{A_i} D_i(p) \) (so the set of trades \( y_i/p_{A_i} \) correspond to an equilibrium of the original Arrow-Debreu exchange market). When such an equilibrium exists, the minimum objective value is 0. \( \square \)

Subject to minimal assumptions (such as Assumption 3), Lemma A.4 shows that an exchange market equilibrium always exists.

**Lemma 6.7.** The so-called convex program of §6.2 is convex and feasible.

**Proof.** Each term \( d_i(z) \ln(p_{A_i}/\hat{\beta}_{i,z}(p)) \) is convex, and the integral or sum of convex functions is convex. Feasibility follows from setting \( y_i = 0 \) for all \( i \) and and \( P_C = 1 \) for all \( C \in \mathcal{C} \). \( \square \)

As an observation, note that [24] requires a technical condition (condition *) to ensure feasibility and that there exists a market equilibrium with non-zero prices.

Informally, we avoid infeasibility here by combining the first and second constraints of the program of [24], and effectively upper bound the \( y_i \) variables through the utility calculation in the \( g_i \) function.

**Lemma 6.8.** The objective value of the convex program is nonnegative.

**Proof.** By construction, for all \( z \), \( \ln \hat{\beta}_{i,z}(p) \leq \ln(z) + \ln(p_{B_i}) \), or alternatively, \( \ln(1/z) \leq -\ln(\hat{\beta}_{i,z}(p)) + \ln(p_{B_i}) \).

Rearranging the last term of the objective gives

\[
\sum_{i \in N} P_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(1/z)dz + \sum_{i \in M} y_i \ln(1/r_i) \leq \sum_{i \in N} P_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(\hat{\beta}_{i,z}(p)) dz + \sum_{i \in M} y_i \ln(\hat{\beta}_{i,r_i}(p)) + \ln(p_{B_i})
\]

Note that \( P_{A_i} D_i^{-1}(y_i/p_{A_i}) d_i(z) dz = \frac{y_i P_{A_i}}{P_{A_i}} = y_i \).

Applying the first program constraint gives the following (note the change in subscripting)

\[
\sum_{i \in N} \left( P_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(p_{B_i}) dz \right) + \sum_{i \in M} y_i \ln(p_{B_i})
\]

\[
= \sum_{C \in \mathcal{C}} \sum_{i : A_i = C} y_i \ln(p_C) = \sum_{C \in \mathcal{C}} \sum_{i : A_i = C} y_i \ln(p_C)
\]

\[
= \sum_{i \in N} \left( P_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(p_{A_i}) dz \right) + \sum_{i \in M} y_i \ln(p_{A_i})
\]
Hence,
\[
\sum_{i \in N} P_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(1/z)dz + \sum_{i \in M} P_{A_i} (y_i/p_{A_i}) \ln(1/r_i)) \\
\leq \sum_{i \in N} P_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) (\ln(p_{A_i}) - \ln(\beta_{i,z}(p))) dz \\
+ \sum_{i \in M} y_i (\ln(p_{A_i}) - \ln(\beta_{i,r_i}(p)))
\]

As \(\beta_{i,z}(p)\) is at most \(p_{A_i}\) and \(y_i \leq p_{A_i} d_i\) for \(i \in M\), the term inside the integral is nonnegative for any \(z\), and thus this summation is upper bounded by
\[
\sum_{i \in N} P_{A_i} \int_0^{\infty} d_i(z) \ln \left( \frac{P_{A_i}}{\beta_{i,z}(p)} \right) dz + \sum_{i \in M} P_{A_i} d_i \ln \left( \frac{P_{A_i}}{\beta_{i,r_i}(p)} \right)
\]
which is the first two terms of the objective. \(\square\)

Lemma 6.9. The objective of the convex program is 0 if and only if \(y_i = p_{A_i} D_i(p_{A_i}/p_{B_i})\) for all \(i\).

Proof. View the objective function as a first (nonnegative) term from which a second (nonnegative) term is subtracted.

Define \(\hat{z}_i = p_{A_i}/p_{B_i}\).

Following the rewrites of the previous proof, the last term of the objective is upper bounded by
\[
\sum_{i \in N} P_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \left( \ln \left( \frac{P_{A_i}}{\beta_{i,z}(p)} \right) - \ln(\beta_{i,z}(p)) \right) dz + \sum_{i \in M} y_i \ln \left( \frac{P_{A_i}}{\beta_{i,r_i}(p)} \right)
\]
For any \(i \in M\), it must be the case that \(y_i \ln(1/r_i) \leq y_i \ln \left( \frac{p_{A_i}}{\beta_{i,r_i}(p)} \right)\) and that \(y_i \ln \left( \frac{p_{A_i}}{\beta_{i,r_i}(p)} \right) \leq p_{A_i} d_i \ln \left( \frac{p_{A_i}}{\beta_{i,r_i}(p)} \right)\).

For \(r_i < \hat{z}_i\), \(p_{B_i}/\beta_{i,r_i}(p) = 1/\hat{z}_i\), and otherwise \(p_{B_i}/\beta_{i,r_i}(p) = 1/r_i\). Hence, the first inequality is tight if and only if either \(r_i \geq \hat{z}_i\) or \(y_i = 0\). In other words, if and only if \(y_i \leq p_{A_i} D_i(p_{A_i}/p_{B_i})\).

Similarly, the second inequality is tight if and only if \(y_i \geq p_{A_i} D_i(p_{A_i}/p_{B_i})\) (either \(p_{A_i}/\beta_{i,r_i}(p) = 1/\hat{z}_i\) or \(y_i = p_{A_i} d_i\)).

If \(r_i < \hat{z}_i\), then \(p_{A_i}/\beta_{i,r_i}(p) = \hat{z}_i/r_i > 1\), and otherwise, \(p_{A_i}/\beta_{i,r_i}(p) = 1\). Therefore, the first inequality in the above equation is tight if and only if \(y_i \leq p_{A_i} D_i(p_{A_i}/p_{B_i})\). Similarly, the second inequality is tight if and only if \(y_i \geq p_{A_i} D_i(p_{A_i}/p_{B_i})\).

Hence, both inequalities are tight if and only if \(y_i = p_{A_i} D_i(p_{A_i}/p_{B_i})\).

An analogous proof holds for \(i \in N\). Note that \(\ln(p_{A_i}/\beta_{i,z}(p))\) is always nonnegative and is strictly positive for \(z < \hat{z}_i\), and \(\ln(\beta_{i,z}(p)) \leq \ln(z) + \ln(p_{B_i})\) is an equality if and only if \(z = \hat{z}_i\).

As such,
\[
p_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(1/z)dz \leq p_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(p_{B_i}/\beta_{i,z}(p))dz
\]
and this equality is tight only if \(y_i \leq p_{A_i} D_i(p)\).

Similarly,
\[
p_{A_i} \int_0^{D_i^{-1}(y_i/p_{A_i})} d_i(z) \ln(p_{A_i}/\beta_{i,z}(p))dz \leq p_{A_i} \int_0^{\infty} d_i(z) \ln(p_{A_i}/\beta_{i,z}(p))dz
\]
and this equality is tight only if \(y_i \geq p_{A_i} D_i(p)\).

Thus, if there exists some \(i\) with \(y_i \neq p_{A_i} D_i(p)\), either the above upper bound of the last term of the objective is not tight, or the upper bound of the last term is strictly lower than the first two terms of the objective. \(\square\)
6.3 Rationality of Convex Program

The original program of [24] is guaranteed to have a rational solution. This program here may not, if the CFMMs can be arbitrarily constructed. However, rational solutions exist when CFMMs belong to a particular class.

**Theorem 6.10.** If, for every CFMM $i$ in the market, the expression $p_A_i D_i(p_A_i/p_B_i)$ is a linear, rational function of $p_A_i$ and $p_B_i$, on the range where $D(i) > 0$, then the convex program has a rational solution.

**Proof.** At an optimal point $(p, y)$, for every CFMM $i$, it must be the case that $p_A_i D_i(p_A_i/p_B_i) = y_i$.

If $D_i$ satisfies the requirements of the theorem, then there is some linear function $q_i$ for which $y_i = q_i(p_A_i, p_B_i)$ when $y_i \geq 0$.

To the set of existing constraints in the convex program, add for each $i$ with $y_i > 0$ the constraint that $y_i = q_i(p_A_i, p_B_i)$. For each $i$ with $y_i = 0$, add the constraints that $y_i = 0$ and that $q_i(p_A_i, p_B_i) \leq \lim_{r \to 0^+} r D(r)$.

This system of constraints is clearly satisfiable, and every point in the system is in fact a market equilibrium (every point satisfies $p_A_i D_i(p_A_i/p_B_i) = y_i$). Each of the constraints is linear and rational, so these constraints define a rational polytope. The extremal points of this polytope must therefore be rational. □

**Example 6.11.** Some natural CFMMs satisfy the condition of Theorem 6.10.

- The density function of the constant product rule with reserves $(A_0, B_0)$ is $\max(0, (pA_0 - B_0)/(2p))$. Substitution gives (for relative exchange rates above the initial spot exchange rate) $p_A_i D_i(p_A_i/p_B_i) = (p_A_i, A_0 - p_B_i B_0)/2$.

- The density function of the constant sum rule $f(a, b) = ra + b$ with reserves $(A_0, B_0)$ is $A_0$ if $p > r$ and 0 otherwise.

However, this convex program cannot always have rational solutions; in fact, there exist simple examples using natural utility functions for which the program has only irrational solutions.

**Example 6.12.** There exists a batch instance containing one CFMM based on the logarithmic market scoring rule and one limit sell offer that only admits irrational equilibria.

Note that it is not required for this example that the LMSR use the natural logarithm. What is required for this example is that the base of the logarithm $b$ and the limit sell offer’s minimum price $p$ be such that $\log_b(p)$ be irrational.

7 CONCLUSION AND OPEN PROBLEMS

Constant Function Market Makers have become some of the most widely used exchange systems in the blockchain ecosystem. Batch trading has been proposed and deployed to combat some shortcomings of decentralised and traditional exchanges. Different implementations in practice have taken substantially different approaches to how these two innovations should interact.

We show here from a minimal set of axioms describing the positive properties of these systems that there is a unique rule for integrating CFMMs into batch trading schemes that preserves the desirable properties of both systems together.

We also study how the CFMM trading functions interact with the efficient computation of market equilibria. Finally, we construct a convex program that computes equilibria on Arrow-Debreu exchange markets where agents have utility for only two assets. When applied to market instances derived from natural CFMM trading functions, the objective of this convex program is smooth, and the program has rational solutions.
7.1 Open Problems

In §6, we build a convex program that integrates 2-asset CFMMs but does not present a way to integrate 3+ asset CFMMs, even for those that satisfy WGS. A convex program that handles 3+ asset CFMMs, perhaps for a narrow class of CFMMs, would be interesting.

In this sense, 2-asset CFMMs appear easier to work with than 3+ asset CFMMs. Rigorously understanding the difference, if there is any, between 2-asset and 3+ asset CFMMs is an interesting line of future work.

From an applied perspective, iterative algorithms like Tâtonnement [21], if implemented naïvely, would require for each demand query an iteration over every CFMM. Ramseyer et al. [47] give a preprocessing step to enable computation of the aggregate response of a group of limit sell offers in logarithmic time; identifying a subclass of CFMM trading functions that admit an analogous preprocessing step or more generally, some way of computing the aggregate demand response of a large group of CFMMs in sublinear time would be of great practical use for batch exchanges using these types of algorithms.

ACKNOWLEDGEMENTS

This research was supported by the Stanford Future of Digital Currency Initiative, the Stanford Center for Blockchain Research, the Office of Naval Research (ONR N00014-19-1-2268), and the Army Research Office (76412CSII).

REFERENCES

[1] Cowswap faq. https://cowswap.exchange/#/faq#how-does-cowswap-determine-prices. Accessed 2/9/2022.
[2] An exchange protocol for the decentralized web. https://docs.gnosis.io/protocol/docs/introduction1 and https://github.com/gnosis/dex-research/blob/8b204510c5d47533bae9ee42bf0498d087f7a78/dfusion/dfusion.v1.pdf.
[3] Uniswap v2 smart contract source. https://github.com/Uniswap/v2-core/blob/1136544ace842fd48ae0b1b939701436598d74075/contracts/UniswapV2Pair.sol#L159, 2020.
[4] Building speedex - a novel design for a scalable decentralized exchange. https://stellar.org/blog/building-speedex-a-novel-design-for-decentralized-exchanges, Nov 2021. Accessed 2/10/2022.
[5] Nyse opening and closing auctions. https://web.archive.org/web/20210708141526/https://www.nyse.com/publicdocs/nyse/markets/nyse/NYSEOpening_n_1_9_13_2021.pdf, 2021.
[6] The penumbra protocol: Sealed bid batch swaps. https://protocol.penumbra.zone/main/zswap/swap.html, 2022.
[7] Hayden Adams, Noah Zinsmeister, and Dan Robinson. Uniswap v2 core. 2020.
[8] Guillermo Angeris and Tarun Chitra. Improved price oracles: Constant function market makers. In Proceedings of the 2nd ACM Conference on Advances in Financial Technologies, pages 80–91, 2020.
[9] Guillermo Angeris, Tarun Chitra, Alex Evans, and Stephen Boyd. Optimal routing for constant function market makers. preprint arXiv:2204.05238, 2022.
[10] Guillermo Angeris, Alex Evans, and Tarun Chitra. Replicating monotonic payoffs without oracles. preprint arXiv:2111.13740, 2021.
[11] Matteo Aquilina, Eric Budish, and Peter O’neill. Quantifying the high-frequency trading “arms race”. The Quarterly Journal of Economics, 137(1):493–564, 2022.
[12] Kenneth J Arrow, Henry D Block, and Leonid Hurwicz. On the stability of the competitive equilibrium, u. Econometrica: Journal of the Econometric Society, pages 82–109, 1959.
[13] Kenneth J Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. Econometrica: Journal of the Econometric Society, pages 265–290, 1954.
[14] Xiaohui Bei, Jugal Garg, and Martin Hoefer. Ascending-price algorithms for unknown markets. ACM Transactions on Algorithms (TALG), 15(3):1–33, 2019.
[15] Eric Budish, Peter Cramton, and John Shim. Implementation details for frequent batch auctions: Slowing down markets to the blink of an eye. American Economic Review, 104(5):418–24, 2014.
[16] Eric Budish, Peter Cramton, and John Shim. The high-frequency trading arms race: Frequent batch auctions as a market design response. The Quarterly Journal of Economics, 130(4):1547–1621, 2015.
[17] Agostino Capponi and Ruijhe Jia. The adoption of blockchain-based decentralized exchanges. preprint arXiv:2103.08842, 2021.
References

[18] Álvaro Cartea, Faycal Drissi, and Marcello Monga. Decentralised finance and automated market making: Predictable loss and optimal liquidity provision. Available at SSRN 4273989, 2022.

[19] Xi Chen, Decheng Dai, Ye Du, and Shang-Hua Teng. Settling the complexity of arrow-debreu equilibria in markets with additively separable utilities. In 2009 50th Annual IEEE Symposium on Foundations of Computer Science, pages 273–282. IEEE, 2009.

[20] Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. The complexity of non-monotone markets. Journal of the ACM (JACM), 64(3):1–56, 2017.

[21] Bruno Codenotti, Benton McCune, and Kasturi Varadarajan. Market equilibrium via the excess demand function. In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, pages 74–83, 2005.

[22] Bruno Codenotti, Srijan V Pemmaraju, and Kasturi R Varadarajan. On the polynomial time computation of equilibria for certain exchange economies. In SODA, volume 5, pages 72–81, 2005.

[23] Philip Daian, Steven Goldfeder, Tyler Kell, Yanqi Li, Xuexuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Front-running in decentralized exchanges, miner extractable value, and consensus instability. In 2020 IEEE Symposium on Security and Privacy (SP), pages 910–927. IEEE, 2020.

[24] Nikhil R Devanur, Jugal Garg, and László A Végh. A rational convex program for linear arrow-debreu markets. ACM Transactions on Economics and Computation (TEAC), 5(1):1–13, 2016.

[25] Nikhil R Devanur and Vijay Vazirani. Extensions of the spending constraint model: Existence and uniqueness of equilibria. In Proceedings of the 4th ACM conference on Electronic commerce, pages 202–203, 2003.

[26] Nikhil R Devanur and Vijay Vazirani. An improved approximation scheme for computing Arrow-Debreu prices for the linear case. In International Conference on Foundations of Software Technology and Theoretical Computer Science, pages 149–155. Springer, 2003.

[27] Ran Duan and Kurt Mehlhorn. A combinatorial polynomial algorithm for the linear Arrow–Debreu market. Information and Computation, 243:112–132, 2015.

[28] Michael Egorov. Stableswap-efficient mechanism for stablecoin liquidity. Retrieved Feb 2021, 2019.

[29] Edmund Eisenberg and David Gale. Consensus of subjective probabilities: The pari-mutual method. The Annals of Mathematical Statistics, 30(1):165–168, 1959.

[30] Alex Evtov, Guillermo Angeris, and Tarun Chitra. Optimal fees for geometric mean market makers. In International Conference on Financial Cryptography and Data Security, pages 65–79. Springer, 2021.

[31] Zhou Fan, Francisco Marmolejo-Cossío, Ben Abshuder, He Sun, Xintong Wang, and David C Parkes. Differential liquidity provision in uniswap v3 and implications for contract design. preprint arXiv:2204.00464, 2022.

[32] Jugal Garg, Edin Hüsić, and László A Végh. Auction algorithms for market equilibrium with weak gross substitute demands and their applications. preprint arXiv:1908.07948, 2019.

[33] Jugal Garg and László A Végh. A strongly polynomial algorithm for linear exchange markets. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pages 54–65, 2019.

[34] Austin Gerig and David Michayluk. Automated liquidity provision and the demise of traditional market making. preprint arXiv:1007.2352, 2010.

[35] Lawrence R Glosten and Paul R Milgrom. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. Journal of Financial Economics, 14(1):71–100, 1985.

[36] Mohak Goyal, Geoffrey Ramseyer, Ashish Goel, and David Mazières. Finding the right curve: Optimal design of constant function market makers. preprint arXiv:2111.02719, 2021.

[37] Robin Hanson. Logarithmic markets coring rules for modular combinatorial information aggregation. The Journal of Prediction Markets, 1(1):3–15, 2007.

[38] Robin Hanson. Logarithmic markets coring rules for modular combinatorial information aggregation. The Journal of Prediction Markets, 1(1):3–15, 2007.

[39] Robin Hanson. Logarithmic markets coring rules for modular combinatorial information aggregation. The Journal of Prediction Markets, 1(1):3–15, 2007.

[40] Jonathan Jove, Geoffrey Ramseyer, and Jay Gerag. Core advancement protocol 45: Speedex: A scalable, parallelizable, and economically efficient digital exchange. preprint arXiv:2212.03340, 2022.

[41] Albert S Kyle. Continuous auctions and insider trading. Econometrica: Journal of the Econometric Society, pages 1315–1335, 1985.

[42] Albert S Kyle. Informed speculation with imperfect competition. The Review of Economic Studies, 56(3):317–355, 1989.

[43] Alfred Lehár, Christine A Parlour, and Marius Zocan. Liquidity fragmentation on decentralized exchanges. Available at SSRN 4267429, 2022.

[44] Jason Milionis, Ciamac C Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing. arXiv preprint arXiv:2008.06046, 2022.

[45] Michael Neuder, Rithvik Rao, Daniel J Moroz, and David C Parkes. Strategic liquidity provision in uniswap v3. preprint arXiv:2106.12033, 2021.

[46] Abraham Othman, David M Pennock, Daniel M Reeves, and Tuomas Sandholm. A practical liquidity-sensitive automated market maker. ACM Transactions on Economics and Computation (TEAC), 1(3):1–25, 2013.

[47] Geoffrey Ramseyer, Ashish Goel, and David Mazières. Speedex: A scalable, parallelizable, and economically efficient digital exchange. preprint arXiv:2111.02719, 2021.

[48] Tom Walther. Multi-token batch auctions with uniform clearing prices - features and models. 2022.

[49] Yinyu Ye. A path to the Arrow–Debreu competitive market equilibrium. Mathematical Programming, 111(1-2):315–348, 2008.
A PROPERTIES OF ARROW-DEBREU EXCHANGE MARKETS

For completeness, we reproduce here some general facts about Arrow-Debreu exchange markets, written in the language of the axioms of §3 when relevant.

Lemma A.1. Suppose that all utility functions \( u(\cdot) \) in a batch exchange satisfy Assumption 1.

Any batch exchange using these utility functions that satisfies Axioms 1, 2, and 3 must also satisfy Axiom 4.

Proof. Let \( u(\cdot) \) be any utility function satisfying the assumptions of the lemma. First, observe that assumptions 1 and 3 imply that the set \( \{ e | u(e) \geq k \} \) is convex and unbounded for any \( k \in \mathbb{R} \).

Next, for any real values \( k_1 < k_2 \) and any set of prices \( p \), let \( e_1 \) maximize \( u(\cdot) \) subject to \( p \cdot e_1 \leq k_1 \) and \( e_1 \geq 0 \), and \( e_2 \) maximize \( u(\cdot) \) subject to \( p \cdot e_2 \leq k_2 \) and \( e_2 \geq 0 \) (if \( u(\cdot) \) is unbounded subject to these constraints, then there cannot exist a set of trades that settles a batch using this utility function at these prices, so Pareto-optimality holds vacuously).

It must be the case that \( u(e_1) < u(e_2) \) (naturally, \( u(e_1) \leq u(e_2) \)). Let \( e_3 \geq 0 \) be some endowment with \( u(e_3) > u(e_2) \) such that \( k_1 = p \cdot e_3 > k_2 \), and let \( e_2' = (e_2 - k_3 + k_2) + (e_1 - k_2 + k_1) \). By assumption 2, it must be the case that \( u(e_2') > u(e_1) \), and because \( e_2' \cdot p = k_2 \), \( u(e_2) \geq u(e_2') \).

Now, consider any batch exchange instance where utility functions satisfy the assumptions of the lemma, endowments \( \{ e_i \} \), and \( \{ x_i \} \) any set of trades that settles the batch subject to Axioms 1 (with valuations \( p \)), 2, and 3.

For any set of additional trades \( \{ y_i \} \) with \( \Sigma_i y_i = 0 \), if the utility function of some agent \( i \) is strictly improved, then it must be the case that \( p \cdot y_i > 0 \). Thus, there must exist \( i_2 \) with \( p \cdot y_{i_2} < 0 \), and as such it must be the case that \( u_{i_2}(e_{i_2} + x_{i_2} - y_{i_2}) < u_{i_2}(e_{i_2} + x_{i_2}) \). Thus, \( \{ y_i \} \) is not Pareto-improving, and thus Axiom 4 is always satisfied.

Lemma A.2. Computing a set of trades that settles a batch exchange instance subject to Axioms 1, 2, and 3 is equivalent to computing an equilibrium of an Arrow-Debreu exchange market with nonzero prices.

Proof. First, consider a batch exchange instance and a set of trades \( \{ x_i \} \) that settles the batch subject to Axioms 1, 2, and 3 (using valuations \( p \)).

Then \( (p, \{ e_i + x_i \}) \) is an equilibrium of the Arrow-Debreu exchange market where each batch participant corresponds to an agent with endowment \( e_i \) and utility \( u_i(\cdot) \) (Axiom 3 implies that \( e_i + x_i \) is a best response for agent \( i \) to prices \( p \), and Axiom 2 ensures that \( \Sigma_i (e_i + x_i) \leq \Sigma_i e_i \) for all assets \( A \).

Next, consider any equilibrium \( (p, \{ y_i \}) \) of this Arrow-Debreu exchange market with strictly nonzero prices. Then the set of trades \( \{ x_i = y_i - e_i \} \) are optimal for each batch participant (satisfying Axiom 3. For each asset \( A \), because \( p \cdot A > 0 \), \( \Sigma_i (y_i) \cdot A = \Sigma_i (e_i) \cdot A \) and thus \( \Sigma_i (x_i) \cdot A = 0 \).

Corollary A.3. A batch exchange can be settled subject to Axioms 1, 2, and 3 precisely when the corresponding Arrow-Debreu exchange market admits an equilibrium with nonzero prices.

Note that Axiom 1 to allow valuations of 0 and Axiom 2 could be relaxed to allow assets to be burned when the valuation is zero (so as to align the axioms more closely to Arrow-Debreu exchange markets).
A.1 Existence of Equilibria with Positive Valuations

Equilibria with positive asset valuations exist when participant utility functions satisfy mild conditions. The following lemma is directly adapted from Arrow and Debreu [13] and Assumption 3 is adapted from condition * of Devanur et al. [24].

Assumption 3. For every asset $\mathcal{A}$ and every set of prices $p$, if $p_{\mathcal{A}} = 0$, then there exists at least one agent who always has positive marginal utility for $\mathcal{A}$ (no matter how much $\mathcal{A}$ the agent purchases).

Lemma A.4. If Assumption 3 holds in some Arrow-Debreu exchange market where every utility function satisfies Assumption 1, then there always exists an equilibrium of that market and every equilibrium has a positive valuation on every asset.

Proof. Let $\hat{e} = (\Sigma_i e_i) + (1, \ldots, 1)$ be a vector of assets strictly larger than the total amount of each asset available in the market, and let $E = \{x \mid 0 \leq x \leq \hat{e}\}$. Let $P$ be the price simplex on $\mathcal{F}$ ($P = \{p \mid \Sigma \mathcal{F} p_{\mathcal{A}} = 1\}$).

Consider the following game. There is one player for each agent in the exchange market and one “market” player, for a combined state space of $E \times \ldots \times E \times P$. Clearly, this state space is compact, convex, and nonempty.

Given the set of prices $p$, each agent player picks a utility-maximizing set of goods $x_i$ subject to resource constraints (specifically, for each agent $i$, $x_i$ maximizes $u_i(\cdot)$ subject to $p \cdot x \leq p \cdot e_i$ and $x_i \in E$), receiving payoff $u_i(x_i)$. The market player chooses a set of prices $\hat{p}$, for payoff $\Sigma_i e_i - x_i \cdot p$.

Define $A_i(p) = \{x \mid x \in E, p \cdot x \leq p \cdot e_i\}$ to be the set of utility-maximizing sets of goods for agent $i$. Quasi-concavity of $u_i(\cdot)$ implies that $A_i(p)$ is convex, $A_i(\cdot)$ cannot be nonempty. It suffices to show that $A_i(\cdot)$ is a continuous function for each agent $i$.

Let $p_1, p_2, \ldots$ and $x_1, x_2, \ldots$ be any sequences of prices and demand responses converting to $p$ and $x$, respectively, with $x_j \in A_i(p_j)$ for all $j \in \mathbb{Z}_+$. Let $r_j = p_j \cdot x_j$ for each $j$. Naturally, the sequence $r_1, r_2, \ldots$ must converge to $r = p \cdot x$, and the sequence $u_i(x_1), u_i(x_2), \ldots$ must converge to $u_i(x)$.

Consider a sequence $\{x'_j\}$ converging to $x'$ with $p_j \cdot x'_j = r_j$ and $x' \in A_i(p)$. It must be the case that $u_i(x'_j) \geq u_i(x'_j')$. Because $u_i(x'_j)$ converges to $u_i(x')$, we must have that $u_i(x) \geq u_i(x')$, so $x \in A_i(p)$ and thus $A_i(\cdot)$ must be continuous.

It follows from Kakutani’s fixed point theorem that there must exist a fixed point of this game; that is, there exists a $p$ and an $x_i$ for each $i$ such that $x_i \in A_i(p)$ and $(\Sigma_i x_i)_{\mathcal{A}} \leq 0$, with the inequality tight if $p_{\mathcal{A}} > 0$ for all assets $\mathcal{A}$.

By Assumption 3, if $p_{\mathcal{A}} = 0$ for some asset $\mathcal{A}$, then there exists an agent $i$ for which every $x_i \in A_i(p)$ has $(x_i)_{\mathcal{A}} = \hat{e}_{\mathcal{A}}$. But then the demand for $\mathcal{A}$ must exceed the available supply, so $(\Sigma_i x_i)_{\mathcal{A}} > 0$, a contradiction. Analogously, it must be the case that for every asset $\mathcal{A}$ and every agent $i$, it must be the case that $(x_i)_{\mathcal{A}} < \hat{e}_{\mathcal{A}}$, and thus (by parts 2 and 3 of Assumption 1) $x_i$ maximizes $u_i(\cdot)$ subject to $x_i \cdot p \leq e_i, x_i \geq 0$ (i.e. without the restriction that $x_i \in E$).

The above proof needs only small modifications if agents are specified by their spending (as in Definition 5.4). Ideally, one would set $A_i(p)$ to be the single vector with $(x_i)_{\mathcal{A}} = \frac{e_i}{\Sigma \mathcal{F}} A_i(p)$. However, this might place $x_i$ outside of $E$, so instead, we define $A_i(\cdot) = \{x_i\}$ as above when $x_i \in E$, and otherwise, cap the amount spent on any good that would take $x_i$ out of $E$ appropriately, and divide the leftover budget to spend on the other goods in equal proportion (repeating as necessary as asset quantities reach their limits in $E$). This modified $A_i(\cdot)$ is clearly a continuous function (so the graph is closed) and each $A_i(\cdot)$ is a nonempty, convex set. Thus, (again, subject to Assumption 3), the lemma holds.
B PROOFS FOR SECTION 3 (AN AXIOMATIC APPROACH TO CFMMS IN BATCH TRADING SCHEMES)

Example 3.1. Consider the following batch trading instance with one CFMM and two limit sell orders (Example 2.3).

- The first limit order sells up to 1 unit of \( A \) for \( B \) with a minimum price of 1 \( B \) per \( A \).
- The second sells up to 3 units of \( B \) in exchange for \( A \), with a minimum price of \( \frac{1}{6} A \) per \( B \).
- The CFMM is a constant-product market maker \( (f(x_A, x_B) = x_A x_B) \) with reserves of 1 unit of \( A \) and 10 units of \( B \).

If this instance follows Axioms 1, 2, and 3, and preserves \( f(\cdot) \) constant, then the CFMM can never make any trades in a batch settlement, despite the fact that were such a CFMM deployed classically, the first limit sell order would strictly prefer trading with the CFMM to not trading, and always strictly prefers to trade with the CFMM than with the second order.

Proof. Let \( \{x_i\} \) be some settlement at batch valuations \( p \) satisfying the axioms stated in the example, and let \( \hat{p} = \frac{P_A}{P_B} \) at

- If \( r < 1 \): the second limit order must strictly purchase (in net) \( A \), but the first does not sell \( A \) (without its trading function value decreasing). Such a settlement would therefore violate Axiom 2.
- If \( r > 6 \): the first limit order sells 1 \( A \) for \( r \) units of \( B \), but the second limit order does not sell any \( B \). Under asset conservation (Axiom 2), the CFMM’s final reserves would be \( (2A, (10 - r)B) \). This leads to the CFMMs trading function value decreasing since \( 2 + (10 - r) < 2 + 4 = 8 \), therefore violating Axiom 5.
- If \( (1 \leq r \leq 6) \): the first limit order sells 1 \( A \) and receives \( r B \), while the second limit order sells 3 \( B \) and receives \( 3/r A \). Due to asset conservation (Axiom 2), the CFMM’s final reserves are \( (1 + 1 - 3/r)A \) and \( (10 - r + 3)B \). By the CFMM’s exact trading constraint we have: \( (2 - 3/r)(13 - r) = 10 \). The only solution in \([1, 6]\) is \( r = 3 \) which corresponds to the CFMM not making any trade.

□

□

C PROOFS FOR SECTION 5 (CFMM TRADING FUNCTION STRUCTURE AND COMPUTATIONAL EFFICIENCY)

Proposition 5.3. The problem of computing equilibria in batches of limit sell offers is solvable in polynomial time, but the problem of computing equilibria in batches that include limit buy and limit sell offers is PPAD-complete.

More generally, exchange market equilibria can be approximated in polynomial time if the utility function of every market participant satisfies WGS. However, the problem of computing equilibria in markets that contain non-monotone utility functions is PPAD-complete.

Proof. The proposition is a restatement of Theorem 7 of Codenotti et al. [21] and Corollary 2.1 of Chen et al. [20]. We lay out the full connection for completeness.

Recall from Example 2.3 that limit sell offers have linear utility functions on two assets — functions of the form \( f(x, y) = \alpha_x x + \alpha_y y \), and from Example 5.2 that limit buy offers correspond to linear utility functions on two assets with a threshold — functions of the form \( f(x, y) = \alpha_x x + \min(\alpha_y y, c) \).

Optimal demand responses of linear utility functions can be approximated easily by a demand oracle that satisfies WGS. Approximate computability in polynomial time follows from Theorem 7 of [21]. Alternatively, one could solve the convex program of [24].

6The approximation is to a \((1 + \epsilon)\) level of accuracy for any \( \epsilon > 0 \) and the natural metric of accuracy given in Definition 1 of [21]. "Polynomial time" means polynomial in the size of the input market instance and \( 1/\epsilon \).
The hardness result is Corollary 2.1 of Chen et al. [20] and follows from their Theorem 7. The theorem follows from the construction of two types of sub-markets—"normalized non-monotone markets" and "price-regulating markets"—and linking them with "single-minded" traders.

Example 2.4 of Chen et al. [20] shows how to construct non-monotone markets using only the above types of utility functions, where each non-monotone market contains only two goods. The price-regulating markets have only linear utility functions. The construction builds one price-regulating market for each non-monotone sub-market, trading the same set (and thus same number, 2) of goods — so again, one can build these markets from limit sell offers (it also builds an extra price-regulating market that also trades only two goods).

The "single-minded" traders sell all of their goods for another good at any price—as though their limit prices were 0. These traders sell either one or two types of goods at once, so they do not exactly correspond to limit sell offers, but it is without loss of generality to replace an agent with a linear utility function that sells two types of goods with two separate agents, each of which uses the same utility function but where each only sells one of the original goods.

Example 5.6.

1. The constant product trading function \( u(x_A, x_B) = x_A x_B \) spends half of its budget on each good.

2. The weighted constant product trading function \( u(x_A, x_B) = x_A^{w_A} x_B^{w_B} \) for positive constants \( w_A, w_B \) spends a fraction of its budget on \( A \)

Proof. The spot exchange rate of the weighted constant product rule \( u(x_A, x_B) = x_A^{w_A} x_B^{w_B} \) is \( p = \frac{w_A x_A}{w_B x_B} \). Therefore, at any batch valuations \((p_A, p_B)\), it must be the case that \( p_A w_B x_A = p_B w_A x_B \). Let the agent’s budget be \( p \cdot x = K \).

Solving for \( p_A x_A \) gives \( K \frac{w_A}{w_B} \).

The case of the constant product rule follows from setting \( w_A = w_B = 1 \).

\[ \square \]

D PROOFS FOR SECTION 6 (A CONVEX PROGRAM FOR 2-ASSET WGS UTILITY FUNCTIONS)

Example 6.11. Some natural CFMMs satisfy the condition of Theorem 6.10.

- The density function of the constant product rule with reserves \((A_0, B_0)\) is \( \max(0, (pA_0 - B_0)/(2p)) \).
  Substitution gives (for relative exchange rates above the initial spot exchange rate) \( pA_1D_1(pA_1/pB_1) = (pA_0A_0 - pB_0B_0)/2 \)
- The density function of the constant sum rule \( f(a, b) = ra + b \) with reserves \((A_0, B_0)\) is \( A_0 \) if \( p > r \) and 0 otherwise.

Proof. (1) Given a batch exchange rate of \( p \) units of \( B \) per \( A \), a CFMM using the constant product rule must make a trade so that its reserves \((A_1, B_1)\) after trading satisfy the following two conditions. Without loss of generality, assume \( p \) is greater than the CFMM’s current spot exchange rate of \( B_0/A_0 \), so the CFMM sells \( A \) to the market and purchases \( B \).

First, the spot exchange rate of the CFMM, \( B_1/A_1 \), should be equal to \( p \). And second, the CFMM must trade at the batch exchange rate, so \((A_0 - A_1)p = (B_1 - B_0)\). Thus \( D(p) = A_0 - A_1 \).

Combining these two, we get

\[ \frac{B_0 + pD(p)}{A_0 - D(p)} = p \]

Solving for \( D(p) \) gives

\[ D(p) = \frac{pA_0 - B_0}{2p} \]

(2) The CFMM sells its entire endowment if and only if the exchange rate is at least the constant \( r \).

\[ \square \]
Example 6.12. There exists a batch instance containing one CFMM based on the logarithmic market scoring rule and one limit sell offer that only admits irrational equilibria.

Proof. Consider a batch instance trading two assets \( A \) and \( B \) that contains one CFMM and one limit sell offer. The CFMM uses the trading rule \( f(a, b) = - (e^{-a} + e^{-b}) + 2 \), with initial reserves \( a_0 = b_0 = 1 \). The sell offer is an offer to sell 100 units of \( A \) in exchange for \( B \), with a minimum price of at least \( \frac{1}{2} B \) per \( A \).

If the final batch exchange rate \( p = p_A/p_B \) is strictly greater than \( \frac{1}{2} \), then this limit sell offer must sell the entirety of its \( A \) to receive at least \( 100p > 50 \) units of \( B \). But the CFMM can only provide at most 1 unit of \( B \).

On the other hand, if the final batch exchange rate is strictly less than \( \frac{1}{2} \), then the limit sell offer will not sell any \( A \) but the CFMM would attempt to buy a nonzero amount of \( A \). Thus, at equilibrium, the batch exchange rate must be equal to \( \frac{1}{2} \).

Let the amount of \( A \) sold to the market by the sell offer be \( x \). Clearly, at equilibrium, \( 0 \leq x \leq 100 \).

Furthermore, the spot exchange rate of the CFMM at equilibrium must be equal to \( \frac{1}{2} \). The spot exchange rate of this CFMM is

\[
\frac{\partial f}{\partial a} = e^{-a} \quad \frac{\partial f}{\partial b} = e^{-b}
\]

Thus, at equilibrium, we must have that

\[
p = e^{-(a_0 - x) + (b_0 - px)}
\]

\[
= e^{-a_0 + b_0} e^{-x - px}
\]

\[
= 1 \cdot e^{-x(1+p)}
\]

Putting this all together gives \( e^{-x \frac{1}{2}} = \frac{1}{2} \). Which gives \( x = \frac{1}{2} \ln(2) \). Hence, no equilibrium in this market can be rational. \( \square \)