Flow Trading[*]

Eric Budish[†] Peter Cramton[‡] Albert S. Kyle[§] Jeongmin Lee[¶]
David Malec[‖]

October 23, 2021
Preliminary Version

Abstract

We propose a new market design for trading financial assets. The design combines three elements: (1) Traders submit persistent piecewise-linear downward-sloping demand curves to trade in shares per second (Kyle and Lee (2017)). (2) Markets clear using frequent batch auctions held at regular intervals, such as once per second (Budish, Cramton, and Shim (2015)). (3) Traders may submit orders to trade portfolios of assets, expressed as arbitrary linear combinations with positive and negative weights, as if they were one asset. Thus, relative to the status quo design: time is discrete instead of continuous, prices and quantities are continuous instead of discrete, and traders can directly trade portfolios. Market clearing quantities and prices are the solution to a quadratic program with linear constraints, constructed by attributing preferences to orders and maximizing imputed gains from trade. Clearing prices and quantities are shown to exist, with the latter unique. Calculating prices is shown to be computationally feasible. Microfoundations for portfolio orders are provided. The market design has several potential benefits relative to the status quo: (1) traders can directly express many common trading demands, which reduces costs and complexity; (2) it reduces the importance of speed; (3) it allows liquidity and price discovery to be easily linked across related assets; and (4) it improves fairness and transparency, as all executable orders trade at the same prices at the same time.

[*] We thank Elizabeth Baldwin, Darrell Duffie, Paul Klemperer, David Parkes, and Marzena Rostek for helpful conversations. Disclosures: Peter Cramton consults on market design and is an academic advisor to Carta on the design of a private equity exchange. Albert S. Kyle has worked as a consultant for various U.S. government agencies on issues related to competition and efficiency in financial markets. He is a non-executive director of a U.S.-based asset management company. The other authors have no relevant or material financial interests that relate to this research. Research support: Budish thanks the University of Chicago Booth School of Business. Cramton thanks the German Research Foundation (DFG, EXC 2126/1-390838866) and the European Research Council (grant 741409).

[†] Professor of Economics, University of Chicago, Booth School of Business.

[‡] Professor of Economics, University of Cologne and University of Maryland.

[§] Professor of Finance, University of Maryland.

[¶] Assistant Professor of Finance, Olin Business School, Washington University in St. Louis.

[‖] Research Scholar, University of Cologne and University of Maryland.

1 Introduction

Description of the current design Current exchanges for trading equities and many other financial assets implement a market design with the following features. Most orders are variations on a standard limit order, such as, “Buy 1000 shares of AAPL at $150.00 per share or better,” which has one maximum quantity and one limit price. The orders are processed continuously, one-at-a-time in sequential order, with incoming “executable” orders matched in whole or in part with “nonexecutable” orders resting in the limit order book. Orders are typically for single securities rather than portfolios of securities. Displayed bids and offers respect a minimum “tick size,” which is typically $0.01 per share for U.S. stocks, and minimum “lot size,” which has historically been 100 shares for most U.S. stocks. In some markets, traded quantities must also respect a minimum tick size and minimum lot size.

This market design makes it costly for investors to implement trading strategies. Since the entire quantity for an order resting in the limit order book is subject to immediate execution against the next incoming executable order, resting limit orders risk being “picked off” or “sniped” by a high-frequency trader acting on new information such as new bids, offers, or trades in the same or related securities. This risk makes it more expensive to provide liquidity and, in turn, increases the cost of accessing liquidity. On exchanges which process older resting limit orders before new ones at the same price, the discrete minimum tick size induces a race for “time priority” to be the first in the limit order book. The economic rents in this race, too, correspond to increased costs for investors. Both sniping races and the race for queue priority also increase the market’s complexity.

Institutional investors trading large quantities face additional unnecessary costs and complexity. Institutional investors often strategically choose to smooth their trading out over time to reduce price impact. Doing this efficiently requires placing and canceling hundreds or thousands of small orders for individual stocks. This requires institutional investors to have access to complex, expensive trading platforms to manage their orders. Similarly, trading strategies that entail trading portfolios (e.g., most forms of active and passive investing) or buying some stocks while selling others based on relative valuations (e.g., factor investing, long-short arbitrage) require constantly canceling and replacing orders for stocks as prices fluctuate. Implementing such strategies efficiently also requires access to complex, expensive trading platforms.

Flow trading To reduce the costs and complexity that the current market design imposes on retail and institutional participants, we propose a new market design called “flow trading.”

Flow trading is a combination of three key elements (Table 1). First, instead of placing orders that define quantities as step functions of price, traders place flow orders that specify quanti-

Flow Trading	Traditional Exchange
Downward-sloping piecewise-linear supply and demand curves for flows	Discontinuous step functions for discrete quantities
Batch auctions once per second	Sequential matching one at a time
Orders for portfolios (linear combinations)	Orders for one asset

Table 1: Comparison of Flow Trading with Traditional Exchange

ties as piecewise-linear, downward-sloping functions of price. Because quantities change continuously as a function of price, piecewise-linear schedules ensure that all executable orders execute and execute at the same price. The quantities define execution rates as flows—"Buy a maximum of one share per second until 1000 shares are bought"—rather than as a discrete quantity change —"Buy a maximum of 1000 shares right now."

Second, instead of executing orders one at a time in sequence, orders are processed in discrete time using batch auctions (i.e., "frequent batch auctions"). Suppose the discrete-time interval is one second. An order to buy at a maximum rate of one share per second will buy one share per batch if fully executable, a fraction of a share per batch if partially executable, or no shares per batch if non-executable ("off the market"). Orders persist over many auctions; an order remains outstanding until either the trader cancels it or a user-defined termination criterion is met, such as the cumulative purchase of 1000 shares.

In these batch auctions, prices and quantities are approximately continuous—tiny fractions of shares can trade each second within a nearly continuous price grid.[1] In the status quo market design, making prices and quantities approximately continuous would cause an explosion of message traffic, with traders constantly canceling and replacing orders to improve their queue position. Prices and quantities are therefore discrete, but this leads to inefficiency and complexity—races for queue position, complexities associated with round lots, etc. Here, the combination of flow orders and discrete-time batching allows for prices and quantities to be approximately continuous without issue. *That is, relative to the status quo, the proposed market design makes time discrete instead of continuous, and prices and quantities continuous instead of discrete.*

Third, instead of orders for a single asset, each order is for a portfolio of assets. A "portfolio" is a user-defined linear combination of assets in which the asset weights can be positive or negative. Portfolio orders allow assets to be either complements or substitutes. If two assets

[1] Quantities could be expressed in nano-shares (billionths of shares) and prices in micro-dollars (millionths of dollars): for example, trade 0.123456789 shares per second at price $50.123456.

in a portfolio have weights with the same sign, the assets are complements in the usual sense that an increase in the price of one asset decreases the quantity demanded of the other. If two assets have opposite weights, the assets are substitutes because an increase in the price of one asset increases the quantity demanded of the other. For example, a pairs trade has a positive weight on the stock being bought and a negative weight on the stock being sold. An order to sell the S&P 500 has negative weights on each of the 500 stocks in the index. A standard limit order has a nonzero weight only on the stock being bought or sold. An order to sell a single asset, which represents an upward-sloping supply curve for the asset, is implemented as an equivalent downward sloping demand curve for a portfolio with a negative weight on the asset sold and zero weight on other assets.

Benefits Flow trading has four types of benefits relative to the status quo.

First, investors can directly express many common trading demands to the market. Investors can easily control the urgency of trade by choosing the maximum flow rate—in effect, the ability to trade at the time-weighted average price (TWAP) is built directly into the market design (Kyle and Lee (2017)). Since trading fast incurs large temporary market impact, some traders may decide to trade more gradually, while those with short-lived information may still want to trade more quickly. Investors can also easily trade portfolios of assets, or execute trades that involve buying some assets while selling others, using a single order specifying the relevant assets and their respective positive or negative weights. As one example, investors can use portfolio orders to effectively construct and customize their own index ETFs. As another example, investors can use portfolio orders to execute a "Buy A, Sell B" pair trade strategy.

In the status quo market design, all of these aspects of trading strategy—controlling urgency, buying or adjusting portfolios, engaging in long-short trades—require access to sophisticated trading platforms. This is expensive for large investors and simply unavailable to many small investors. Here, these trading tools are in effect built directly into the market design, reducing both costs and complexity.

Second, the market design reduces the importance of speed. As in Budish, Cramton and Shim (2015), the batch processing means that being δ faster than another participant, if the batch interval is τ, is only relevant with likelihood $\frac{\delta}{\tau}$. For example, being 100 microseconds faster, if the batch interval is 1 second, is only relevant with likelihood $\frac{1}{10000}$. Moreover, flow trading makes the executed quantity proportional to the length of time. This means that even when new information arrives just before the next batch auction, so that regular traders are vulnerable to sniping, competition among fast traders will quickly adjust prices, and the actual quantity executed at unfavorable prices will remain small. In addition to reducing sniping, flow trading also reduces the race for queue position, since there is no longer a need to have a discrete price grid to manage messaging costs. See Li, Wang, and Ye (2021) for a model of trading under the status quo market design that combines sniping races and races for queue position, and shows how the rents in both kinds of races are ultimately paid by non-HFT market participants.

Third, the market design makes it easier for market participants to provide liquidity across correlated assets, and helps link price discovery across correlated assets. Suppose A and B are highly correlated assets. In the continuous market, a change in the price of one asset can lead to a sniping race in the other asset—this adds to the expense of providing liquidity. Under flow trading, a market participant can directly provide liquidity in the pairs trades “Buy A, Sell B” and “Sell A, Buy B” (indeed, the latter is just an offer to sell the former). This means that even if an investor arrives wanting to buy just A, their trade can be automatically incorporated into the clearing prices of both A and B. There need not be a sniping race in asset B, nor is there any “correlation breakdown” of prices between A and B (Budish, Cramton, and Shim (2015)). The pairs trade order is like a string that ties the correlated assets’ prices together, maintaining their underlying economic pricing relationships.

Fourth, the new market design improves transparency and fairness. The key feature is that all orders that are executable at the clearing prices are executed, either at their full rate or a partial rate depending on the order’s pricing parameters, and all orders that execute for a given asset receive the same pricing for that asset. This allows, for example, a retail investor who trades 100 shares over a minute to infer the appropriate execution rate on their order from publicly announced market clearing prices exactly. Similarly, an institutional investor trading a sophisticated portfolio can confirm directly that they received the correct execution. This perhaps should not sound radical, but it is a major transparency improvement over the current market design, where checking whether one’s order received appropriate execution is very difficult (see Tyc (2014)).

Having mentioned these potential benefits, we add an important caveat, which is that flow trading is *not* designed to mitigate market failures related to market power or private information. Market participants still must think strategically about how to trade on private information and manage their price impact, just as in the status quo market design.

Technical Foundations We provide three sets of technical results: on existence and uniqueness of market-clearing prices and quantities; on computability of these prices and quantities; and results that provide micro-foundations for the bidding language.

To prove existence of equilibrium prices and quantities, we transform the problem into a well-understood quadratic optimization problem with linear constraints. To do so, we first formulate a quasi-linear quadratic utility function for each order by interpreting the order as an expression of preferences defining a linear marginal utility curve over the range where it is partially executable. The sum of these utility functions creates a concave planner objective function. The restrictions that each order must execute at a rate between zero and its maximum rate (e.g., one share per second) are linear inequality constraints. Market clearing defines linear equality constraints for each asset. Zero trade is feasible, i.e., satisfies both sets of constraints. This setup allows us to use known results from the convex optimization literature to prove existence of unique equilibrium quantities.

Equilibrium prices are found as Lagrange multipliers of the primal problem. Regardless of whether assets are complements or substitutes, market-clearing prices exist because our language imposes downward sloping demand curves on all user-defined portfolios. (We discuss the connection to other existence and non-existence results in the literature in the next subsection). Prices, however, may be non-unique when there are no partially executable orders from which unique prices can be inferred. For example, when there is only one order to buy or sell some asset, the market clearing quantity must be zero, but any price at which the order is non-executable clears the market. Prices can easily be made unique by introducing a tie-breaking rule, such as selecting the clearing prices closest to the prices from the last auction.

To show computational feasibility of the market design, we start by showing our problem has a structure such that the gradient method (i.e., tatonnement) is guaranteed to converge. This proves that our problem is computationally simpler than some cases of finding competitive equilibrium prices (Scarf and Hansen (1973)), as the reader will anticipate from the quadratic-programming setup described just above. It is well known, however, that the gradient method can be slow and inaccurate for problems with this structure. We therefore add to the market design that the exchange itself can serve as a "market maker of last resort". Formally, the exchange is willing to buy or sell an epsilon amount of any portfolio at clearing prices. This allows us to use interior point methods, which are known to be much faster and more accurate than the gradient method. Without the exchange as market maker, we know that zero trade is feasible but it is not strictly on the interior of the constraint set; with the exchange as market maker, we can easily find a feasible point strictly on the interior, from which the algorithm can be initialized.

We provide computational proof-of-concept by calculating market clearing prices for a simulated order book using our own implementation of a public-domain interior-point method on an ordinary office workstation. In a market with 30,000 orders and 500 assets, with parameters chosen to try to make the problem difficult, our algorithm calculates prices in about 0.30 seconds. If the number of assets exceeds 2000, the computation time approaches 1.00 second with the same number of orders. If the number of orders increases to 1,000,000, computation time approaches 10 seconds with 500 assets. Conceptually, our goalpost for the computational exercise is to suggest that serious computing power can solve a practical problem of realistic size in less than one second, not just to illustrate the solution to problem in P and not NP.

We provide a stylized micro-foundation for portfolio orders. Portfolio orders cannot express arbitrary preferences. Indeed, with wealth effects, demand schedules may slope upward; such demands cannot be expressed in our language because we require demand schedules to be downward sloping. For a "CARA–normal" investor (with exponential utility or constant absolute risk aversion and subjective beliefs that liquidation values are normally distributed), the demands for assets are linear functions of the asset's own price and the prices of other assets. Such demands cannot be implemented with standard limit orders due to the dependence of demand on prices for other assets. We show that, by rotating the assets in portfolios in a specific manner, such demands can be implemented with downward-sloping portfolio orders consistent with our proposal. In general, implementing N asset demands requires N portfolio orders. If traders believe that assets have a factor structure of rank $K < N$, they can implement the optimum with only K portfolio orders, which may be practically appealing.

1.1 Related Literature

The key conceptual ideas behind this paper's market design proposal—piecewise-linear downward-sloping demand schedules, portfolios as linear combinations of assets, general equilibrium theory, quadratic programming, batch auctions, reducing temporary price impact by trading slowly—are well-understood by researchers in economics and finance. At some level, our contribution is to combine these ideas into a coherent and practical market design for trading financial assets such as stocks, bonds, and futures contracts.

More specifically, our paper builds closely on Kyle and Lee (2017) and Budish, Cramton, and Shim (2015). Kyle and Lee (2017) propose downward sloping, piecewise-linear flow orders for individual assets ("continuously scaled limit orders"). Budish, Cramton, and Shim (2015) propose frequent batch auctions as a market design for financial exchanges. Combined, these two market design ideas yield a market design for financial assets in which time is discrete instead of continuous, and prices and quantities are continuous instead of discrete; this paper is the first to point that out, but the point may be obvious. The third ingredient of the market design proposal, portfolio orders, is a novel contribution. More precisely, the broad idea of bidding for financial portfolios instead of individual assets is obvious from the combinatorial auctions literature, but our specific language for portfolio bidding is novel, and different potential ways of representing preferences for portfolios might not yield the existence and computability results we obtain here.

Another closely-related body of work is Li, Wang, and Ye (2021), Chao, Yao, and Ye (2019), Chao, Yao, and Ye (2017) and Yao and Ye (2018). This research highlights the complexities created by tick-size constraints in modern markets, and associates tick-size constraints with an important aspect of high-frequency trading, the race for queue position. As emphasized earlier, our market design makes time discrete (in line with Budish, Cramton, and Shim (2015)) and prices continuous.

Sophisticated expression of preferences over multiple objects is a theme in the market design literature more broadly. Research on this topic has straddled computer science, economics, and operations research (Lahaie and Parkes (2004); Sandholm and Boutilier (2006); Milgrom (2009); Klemperer (2010); Vohra (2011); Bichler (2017); Cramton (2017); Budish, Cachon, Kessler, and Othman (2017); Parkes and Seuken (2018); Budish and Kessler (forthcoming)). This literature has mostly focused on indivisible-goods combinatorial allocation problems, such as spectrum auctions. Relative to this burgeoning literature, our contribution is our proposed language for portfolio orders, which treats all goods as perfectly divisible, and allows complementarities and substitutabilities only to the extent that they can be expressed with linear portfolio weights. This language is simple enough to obtain strong existence and computational results, while being expressive enough to capture many important use cases in financial markets.

The idea that optimal trading strategies involve flow trading to reduce temporary price impact costs, even when prices and quantities are continuous, emerges as an equilibrium result in game-theoretic models of rationally-optimizing strategic traders. Black (1971) conjectures that more urgent execution of large orders incurs greater price impact costs. In the context of a continuous-time model of information-based trading among overconfident and privately informed traders, Kyle, Obizhaeva, and Wang (2018) describe an equilibrium in which exponential utility and normal distribution imply all traders optimally submit linear flow strategies. In discrete-time models with trading motivated by private values or endowment shocks, Vayanos (1999) and Du and Zhu (2017) derive optimal trading strategies in which quantities are linear functions of price and inventories become differentiable functions of time in the limit as the time interval between auctions becomes zero.

A growing literature studies the implications of the status-quo market-design requirement that orders to trade an asset to be contingent only on the asset's own price and not on the price of other assets. In a competitive framework, Cespa (2004) studies price efficiency implications when traders instead can make their demands for a given asset contingent not only on the asset's own price but also on other asset prices. The more recent literature emphasizes the importance of strategic trading and price impact. Rostek and Yoon (2020*b*) and Wittwer (2021) find that such fully contingent demand can either increase or decrease welfare depending on market characteristics such as the size of the market and the correlation across assets. Rostek and Yoon (2020*c*) show that the welfare implications of introducing a new synthetic asset, like a portfolio of original assets, depend on price impact and symmetry across traders and assets.

Chen and Duffie (2021) show that trading the same asset in multiple fragmented markets can improve welfare.

Researchers have also investigated the welfare implications of market design when information asymmetries and strategic trading are both important. Rostek and Yoon (2020*a*) survey the literature on strategic trading; see Kyle (1985, 1989) and Klemperer and Meyer (1989) for some early contributions. Duffie and Zhu (2017) examine a specific model with welfare improvement when the market design is based on "size discovery," in which an auctioneer announces prices and traders indicate quantities they are willing to trade at the specified price. Zhang (2020) proposes to tax traders who take liquidity and subsidize traders who provide liquidity. There is also an older proposal for "sunshine trading," in which traders transparently announce quantities before the auction is held to mitigate adverse selection (Wunsch (1986)).

Relationship to General Equilibrium Theory Readers familiar with the standard treatment of general equilibrium theory will notice differences in our approach to existence and uniqueness. Mas-Colell, Whinston, and Green (1995, Chapter 17) ("MWG") is a reference for the standard treatment, descending from Arrow and Debreu (1954) and McKenzie (1959). This standard approach uses fixed-point theorems to derive existence results for general convex preferences which include income effects. Actually finding the fixed point is known to often be computationally intractable (Scarf and Hansen (1973); Daskalakis, Goldberg, and Papadimitriou (2009); Budish, Cachon, Kessler, and Othman (2017)). By contrast, our market design approach focuses on a language for preferences that yields existence and uniqueness within a computationally tractable framework.

There are three main differences with the standard treatment, as explicated in MWG.

First, the setting and assumptions are different.

1. While MWG define preferences for the entire positive orthant, our model defines preferences for a given portfolio on the line segment $(0, q)$, representing partial execution of an order to buy the portfolio. The portfolio can be a short position. By defining utility to be minus-infinity off the line segment, we preserve convexity over a larger space, but we lose continuity.

2. While MWG allow general preferences that allow income effects, we assume quasi-linear utility functions of the form $u(\mathbf{x}) - \boldsymbol{\pi}^\top \mathbf{x}$, which do not have income effects.

3. While MWG require strongly monotone preferences and strictly positive prices, our preferences are not strongly monotone and prices can be negative. Individual assets can be "goods" or "bads". Moreover, it may be difficult to make preferences monotone, even over

the restricted domain of agents' demands, because there is no natural "up" direction for the legs of a pairs trade.

Second, the technique to prove the existence of equilibrium is distinct. While MWG relies on Kakutani's fixed-point theorem, we use quadratic programming.

Third, while equilibrium may not be unique in MWG, we have uniqueness up to a convex set. This results from using quasi-linear utility, which makes the second derivative of the planner's objective function negative (semi) definite, and this guarantees that all equilibria must lie in a convex set. In our framework, substitutes and complements do not matter for existence or uniqueness, since the matrix is negative semi-definite anyway, but substitutes and complements may matter for computational performance.

Relationship to the Indivisible Goods Literature Our assumptions are in some respects more similar to assumptions made in the literature on indivisible goods, which typically uses quasi-linear utility.

A classic reference is Kelso Jr and Crawford (1982), who show that competitive equilibrium is guaranteed to exist in an indivisible goods setting under a substitutes condition. There have been many different variations of the Kelso-Crawford substitutes condition defined in the literature; see Gul and Stacchetti (1999); Milgrom (2000); Hatfield and Milgrom (2005); Ostrovsky (2008); Hatfield et al. (2013). Hatfield et al. (2019) discusses the relationship among many of these criteria and provides a maximum domain result for existence.

Baldwin and Klemperer (2019), on the other hand, use tropical geometry to show that existence can be obtained not only when indivisible goods are substitutes but also in some cases when they are complements. For example, left-shoes and right-shoes are clearly complements, but prices for shoes may nevertheless be guaranteed to exist if all agents' preferences regard them as complements in ways that enable the application of the Baldwin and Klemperer (2019) existence theorems. For example, if all agents purchase shoes as pairs, and no agents regard left shoes and right shoes as substitutes for each other, prices are guaranteed to exist.

Unlike in Baldwin and Klemperer (2019), or in most of the indivisible-goods substitutes literature, we obtain existence for any preferences expressible in our language. This stronger existence result relies on our treatment of all assets as perfectly divisible (avoiding the potential difficulties of exact market-clearing when there are indivisibilities), and—as noted above in the discussion of the relationship to general equilibrium theory—the restriction that preferences are only defined for each portfolio on a line segment exactly corresponding to those portfolio weights, as opposed to preferences being well defined on a richer consumption space.

Two other papers in the indivisible goods literature that stand out as especially related to ours are Klemperer (2010), which proposes the product-mix auction, and Milgrom (2009), which proposes the assignment auction (see also Demange, Gale, and Sotomayor (1986)). Both papers describe multi-object auction designs that use linear preference languages and are motivated in part by financial applications—Klemperer's auction, in particular, was designed for the Bank of England to purchase toxic financial assets during the financial crisis. Technically, the key difference is the preference language. In our design, participants bid for portfolios of assets—e.g., buy a portfolio in which the ratio of AMZN:GOOG is fixed 1:1, at rate up to 1 portfolio unit per second, up to a limit price of $5000. In Klemperer's and Milgrom's designs, participants express preferences over substitutable assets—e.g., I value AMZN at $3000 per share and GOOG at $2000 per share, buy one share of whichever asset gives me greater surplus at the realized prices. This difference in language then drives differences in existence and uniqueness results. The papers also have different intended use cases. We have in mind near-continuous trading of financial assets, in which users trade portfolios in flows. Klemperer's and Milgrom's designs are intended for more of a one-shot, high-value allocation—e.g., a high-value auction for toxic assets during the financial crisis, or a spectrum auction. This difference in intended use case lies behind the difference in the proposed languages.

Structure of the paper The rest of the paper is structured as follows. Section 2 describes flow orders for portfolios. Section 3 discusses the existence and uniqueness of market clearing prices and quantities. Section 4 provides a characterization of equilibrium, discusses optimization approaches, and shows computational feasibility of our proposal. Section 5 provides a microfoundation for portfolio orders. Section 6 discusses implementation and policy issues. Section 7 concludes.

2 Flow Orders for Portfolios

2.1 Formal Definition of Flow Orders

Traditional limit orders consist of a price, quantity, and direction of trade for a single symbol. For example, buy 1000 shares of AAPL at $150.00 per share. The order implicitly defines a stepwise demand curve, with full demand (i.e., 1000 shares) at any price weakly better than the limit, and zero demand at any price strictly worse than the limit.

Flow orders depart from traditional limit orders in 3 ways:

1. Orders are for portfolios of assets instead of individual assets. A portfolio is defined by a vector of weights, $\mathbf{w}_i := (w_{i1}, \ldots, w_{iN})^\top$, where i identifies the order, N denotes the number of assets in the market, and $w_{in} \in \mathbb{R}$ denotes the portfolio weight of asset n in order i. A

strictly positive weight denotes buying the asset, a strictly negative weight denotes selling the asset, and a zero weight denotes that that the asset is not a part of that portfolio.

2. Instead of step-wise demand, flow orders describe piecewise-linear downward-sloping demands. The user specifies two prices, $p_i^L < p_i^H$. The flow order interprets p_i^L as a demand to buy the portfolio in full quantity at prices weakly lower than p_i^L, and interprets p_i^H as indicating zero demand for the portfolio at prices weakly higher than p_i^H. Then, in the interval $[p_i^L, p_i^H]$, the flow order linearly reduces the quantity demanded from full quantity at p_i^L to zero quantity at p_i^H.[2] Note that we use the phrase "buy the portfolio" to include the case of selling assets—in our language, selling an asset is buying a portfolio with a negative weight on the asset at a negative price (i.e., receiving a transfer). We will clarify this point, which we acknowledge is potentially confusing, in detail below.

3. Quantities are expressed as flows per batch interval, up to a total quantity limit. For each order i, the user specifies two quantity parameters, $q_i > 0$ and $Q_i^{\max} > 0$, expressing their demand to buy up to quantity q_i of the portfolio per batch interval, up to a cumulative total purchased quantity of $Q_i^{\max}$. Instead of requiring that quantities express a demand to trade immediately (1000 shares right now!) the user can tune their urgency to trade.

Thus, a flow order is described by the tuple $(\mathbf{w}_i, p_i^L, p_i^H, q_i, Q_i^{\max})$.[3]

Next we formally define a flow order's demand within a batch auction. Assume for now that the order's cumulative purchased quantity is not within q_i of $Q_i^{\max}$, so that the order can purchase its full quantity q_i in the next batch without exceeding $Q_i^{\max}$.[4] Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)^\top$ denote the column vector of market prices of all assets $n = 1, \ldots, N$. The market price for the portfolio defined by the weight vector $\mathbf{w}_i$ is the inner product

$$p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi} := \sum_{n=1}^{N} w_{in} \pi_n. \tag{1}$$

Order i's demand per batch auction, which we call its "flow demand", is the downward-sloping

[2] In a traditional limit order at price p, the implied demand is the full quantity at prices weakly better than p and zero quantity at prices strictly worse than p. In our language, these two implications of the traditional limit price are split into two separate parameters: demand in full at prices weakly better than p_i^L, and demand zero at prices weakly worse than p_i^H.

[3] Throughout this paper, we use a lower-case bold font to denote vectors, an upper-case bold font to denote matrices, a subscript i to denote orders, and a subscript n to denote assets.

[4] In the case where the order's cumulative purchased quantity, say Q_i^t, is within q_i of the limit $Q_i^{\max}$, replace q_i with the remaining quantity demanded $Q_i^{\max} - Q_i^t$. This is a simple way to avoid overshooting. Another possible approach is to take the demand curve implied by the original q_i but truncate it above so that demand never exceeds $Q_i^{\max} - Q_i^t$.

linear function of the portfolio price $p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi}$ defined by:

$$D_i\left(p_i \,\middle|\, \mathbf{w}_i, q_i, p_i^L, p_i^H\right) = q_i \operatorname{trunc}\left(\frac{p_i^H - p_i}{p_i^H - p_i^L}\right), \qquad \text{where} \qquad \operatorname{trunc}(z) := \begin{cases} 1, & \text{for } z \geq 1 \\ z, & \text{for } 0 < z < 1 \\ 0, & \text{for } z \leq 0 \end{cases}. \tag{2}$$

Notice how the rate at which order i buys the portfolio depends on both the order's quantity limit q_i and where the price for the portfolio is relative to the order's price parameters p_i^L and p_i^H. If the portfolio price p_i is less than or equal to p_i^L, the order is "fully executable" and the portfolio is bought at the maximum rate q_i. If the portfolio price p_i is higher than p_i^H, then the order is "nonexecutable" and does not buy at all. If the portfolio price is somewhere between p_i^H and p_i^L, then the order is "partially executable" and buys at the rate determined by linear interpolation between the two price parameters.

Buying vs. Selling This formulation treats "selling" an asset as buying a portfolio with a negative weight on that asset at a negative price. This not only generates compact notation for representing both buying and selling but also emphasizes a symmetry between buying and selling which will be important for understanding how market clearing works. General equilibrium theory often uses this idea that an upward sloping supply curve for positive quantities is equivalent to a downward sloping demand curve for negative quantities.

Whether buying or selling, we have $p_i^L < p_i^H$ and demand defined according to equation (2). However, when selling, both p_i^L and p_i^H are negative. For example, an order to sell XYZ in full at price \$42.00 or higher, with the sell rate declining linearly to zero at price \$41.00, would be encoded with $p_i^L = -\$42.00$ and $p_i^H = -\$41.00$. There are two equivalent ways to remember this. First, think of p_i^L as analogous to the price limit in a traditional limit order (willing to trade in full at this price or better), with demand then declining linearly to zero in the interval $[p_i^L, p_i^H]$. Alternatively, think of p_i^H as the price at which the trader is exactly indifferent between trading and not. Then, as the price improves from p_i^H, the trader's quantity demanded increases linearly, up to a maximum quantity of q_i when the price reaches p_i^L or better.

See Figure 1 for an illustration of buying and selling.

Last, note that if a portfolio has both positive and negative weights, there may not be a natural buying versus selling direction to the order. The trader is always "buying the portfolio" under our approach, but whether their pricing parameters p_i^L and p_i^H are positive or negative will depend on the weighted valuations of the assets in the portfolio.

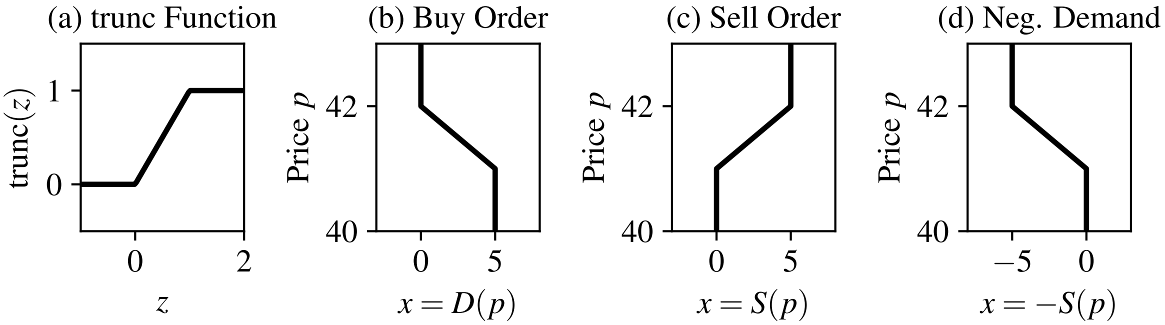

Figure 1: Plots of (a) the function trunc(z); (b) a single buy order, with pricing parameters $p_i^L = \$41.00$ and $p_i^H = \$42.00$, and maximum flow demand of $q_i = 5.00$ portfolio units per batch auction; (c) a single sell order, initially plotted as an upward-sloping supply curve with one upward-sloping linear segment, and (d) the same sell order, now plotted as a downward-sloping demand for negative quantities, which is our treatment here. The pricing parameters for the sell order are $p_i^L = -\$42.00$ and $p_i^H = -\$41.00$, with maximum flow demand of $q_i = 5.00$ portfolio units per batch auction. The figures for buy and sell orders are plotted with flow quantity on the horizontal axis and price on the vertical axis.

Additional Technical Remarks on the Formulation We make two additional technical remarks on this formulation.

First, observe that while the above demand function (2) has just a single downward-sloping segment, the user can define an arbitrary piecewise-linear downward-sloping demand function for a given portfolio by using multiple flow orders.

Second, order specification using the tuple of parameters tuple $(\mathbf{w}_i, p_i^L, p_i^H, q_i, Q_i^{\max})$ contains an intentional redundancy of notation. Buying a portfolio containing one share of a single stock at a rate of two portfolio units per batch auction is equivalent to buying a portfolio containing two shares of the same stock at a rate of one portfolio unit per batch auction. More generally, for some parameter $\alpha > 0$, changing the order parameters from $(\mathbf{w}_i, p_i^L, p_i^H, q_i, Q_i^{\max})$ to $(\alpha\mathbf{w}_i, \alpha p_i^L, \alpha p_i^H, q_i/\alpha, Q_i^{\max}/\alpha)$ has no effect on the trade rates for each asset as a function of asset prices.

Proxy Instructions For Orders Over Time As in the traditional market design, users may modify or cancel their flow orders at any moment in time throughout the trading day. Additionally, users may want to specify what we will refer to as "proxy instructions" that modify or cancel their orders under specified contingencies.

The parameter $Q_i^{\max}$ is a simple example of such a proxy instruction: cancel the order from the market once the cumulative total quantity $Q_i^{\max}$ has been reached. Another simple example is time-in-force instructions, such as "good for day" or good for some other user-specified period of time. In principle, the exchange could provide more complex examples, such as allowing an order's pricing parameters to vary dynamically over time as a function of recent prices ("Ensure that my order's price impact is never more than ten basis points"), or allowing an order's quantity parameter to vary over time ("Reduce this order's flow quantity if I am averaging above ten percent of trading volume"). We will not discuss such complex order contingencies in this paper.

2.2 Key Examples

We give several key examples to illustrate the flexibility of portfolio orders.

1. Standard limit order.

   A standard limit order expresses preferences to buy or sell a fixed quantity of one asset at one limit price. A flow order can be specified to approximate a standard limit order. First, when only one weight w_n is nonzero, the order is a simple order to buy one asset if the weight is positive or to sell one asset if the weight is negative. Second, the maximum rate q_i can be set to equal the quantity the trader wants to buy or sell, $Q_i^{\max}$ (say 1000 shares per batch auction). If fully executable, this order would purchase 1000 shares at the next batch auction. Third, the price parameters can be set so that p_i^L corresponds to the intended limit price, and p_i^H is as close as is allowed to p_i^L. Theoretically, we obtain a standard limit order in the limit as $p_i^H \to p_i^{L+}$.

2. Time-weighted average price (TWAP) order.

   In the traditional market design, a market order executes immediately at the clearing price. The analog here is a time-weighted average price (TWAP) order. The user specifies a price parameter p_i^L that is sufficiently aggressive relative to recent prices that it is esssentially guaranteed to execute.[5] Then, the user will trade quantity q_i of the portfolio every batch auction until their quantity limit is achieved, i.e., they will trade at the TWAP over this time period.

3. Pairs trades.

   A pairs trade can be executed by specifying a portfolio weight vector $\mathbf{w}_i$ with one strictly positive entry, one strictly negative entry, and the rest zeros.

[5] In the traditional formulation of a market order, one thinks of the limit price as ∞ if buying and as 0 if selling. The 0 for selling implicitly encodes that assets are "goods" that can always be sold at a weakly positive price. Here, if the order is for a portfolio with both positive and negative weights, it is not automatic from the order itself whether the portfolio is a "good" that should always trade at a positive price or a "bad" that should trade at a negative price. Either way, the trader can guarantee execution by specifying p_i^L sufficiently large, but they may not wish to do that.

4. Portfolio trades.

   A portfolio trade can be executed by specifying a portfolio weight vector $\mathbf{w}_i$ with either all entries weakly positive (if buying the portfolio) or all entries weakly negative (if selling the portfolio). The assets whose weights are strictly positive or strictly negative comprise the portfolio.

   We note that traders can construct and trade their own index portfolios. For example, an order to buy the S&P 500 has positive weights on each stock in the S&P 500 index, with weights proportional to S&P 500 weights and zero weight on stocks not in the S&P 500 index. An order to sell an index has negative weights on all stocks in the index. Traders can easily customize the index portfolios by adjusting portfolio weights, over-weighting desirable assets and under-weighting the others.

5. General long-short strategies.

   A general long-short strategy combines the previous two cases: multiple positive entries and multiple negative entries.

6. Market making strategies.

   A trader can engage in market making—whether for a single asset, a pairs trade, a portfolio trade, or a general long-short strategy—by using two orders with opposite-signed weights and price parameters. For example, a market maker who is willing to buy portfolio $\mathbf{w}_i$ in full at 41.00 and sell it in full at 42.00, could use orders like

   - Buy leg: weights $\mathbf{w}_i$, price parameters $p_i^L = \$41.00$, $p_i^H = \$41.25$
   - Sell leg: weights -$\mathbf{w}_i$, price parameters $p_i^L = -\$42.00$, $p_i^H = -\$41.75$

2.3 Limitations of the Language

We note several important limitations of the language for representing trading demands.

First, trading demands are only defined at exactly the ratio of portfolio weights specified in the order. If an order specifies it wants to buy assets A and B at a ratio of 2:1, the order contains no information about the trader's willingness to trade at, say, a ratio of 2.2:1 or 1.8:1. This restriction relative to traditional consumer theory, where preferences are typically defined on the whole positive orthant, is key to our method of existence proof (below in Section 3.3). However, this restriction may mean that a trader has to modify their portfolio weights if prices change, either themselves or via proxy instructions. This is a limitation.

Second, trading demands are linear within each order. In principle, we could replace the linear trunc function with the flexibility to specify an arbitrary downward-sloping function on the interval of prices $[p_i^L, p_i^H]$. However, our existence proof and computational results do take advantage of this linearity. We view the linearity restriction as less important a limitation than some of the others, because arbitrary downward-sloping functions can be approximated, if needed, with a set of linear orders.

Third, the language does not allow for indivisibilities. Most importantly, a user cannot specify a minimum transaction quantity per batch, only a maximum. So, for example, an order cannot be "fill or kill", or "at least 100 shares per batch, otherwise stay out". That said, a user may be able to approximate such preferences with marketable orders if prices are continuous enough.

Last, the language does not allow for in-order contingencies. This includes cases like "buy A if the price of B is high enough" or "buy whichever of A or B gives me more surplus given my valuations". This latter kind of preference expression is analyzed in Demange, Gale, and Sotomayor (1986) and is present in market design proposals of Klemperer (2010) and Milgrom (2009). As with indivisibilities, a user may be able to approximate such preferences with marketable orders if prices are continuous enough.

3 Market Clearing Prices and Quantities

Now we turn our attention to the exchange's problem of finding clearing prices and quantities.

3.1 Definition of Market Clearing

To define market clearing we need to convert individual traders' demand curves for portfolios as a function of portfolio prices into a market demand curve for assets as a function of asset prices. For each portfolio i, first replace the portfolio price p_i by the weighted vector of asset prices, using $p_i = \boldsymbol{\pi}^\top \mathbf{w}_i$, then convert the demand for portfolio units $D_i(\boldsymbol{\pi}^\top \mathbf{w}_i)$ into the demand for individual assets by multiplying by the portfolio weights $\mathbf{w}_i$. Next, sum up the demand for assets across all orders i to obtain the market net excess demand curve for assets as a function of asset prices:

$$D(\boldsymbol{\pi}) := \sum_{i=1}^{I} D_i \left(\boldsymbol{\pi}^\top \mathbf{w}_i \,\middle|\, \mathbf{w}_i, q_i, p_i^L, p_i^H \right) \mathbf{w}_i. \tag{3}$$

The function $\mathbf{q} = D(\boldsymbol{\pi})$ maps asset price vectors $\boldsymbol{\pi} \in \mathbb{R}^N$ to net asset quantity vectors $\mathbf{q} \in \mathbb{R}^N$. The market clearing equation $D(\boldsymbol{\pi}) = \mathbf{0}$ defines N equations in N unknowns.

The exchange finds the clearing prices and allocations by solving for a price vector $\boldsymbol{\pi}$ such that net excess demand for each asset is zero:

$$D(\boldsymbol{\pi}) = \mathbf{0}. \tag{4}$$

Once the clearing prices are determined, the trading rate for the assets defined by each individual portfolio order is uniquely determined by the portfolio demand curve $x_i = D_i(\boldsymbol{\pi}^{\top}\mathbf{w}_i)$.

The key difference between the market-clearing mechanism of our design and that of the conventional exchange is that clearing prices must be calculated for all assets simultaneously. Specifically, to find clearing prices for assets, the exchange needs to calculate for each vector of N asset prices the aggregate demand for each asset by summing across all orders the product of the buy rate of the order's portfolio and the portfolio weight of the asset. Of course, for many orders, the portfolio weight of an arbitrary asset is likely to be zero. Since orders are functions of the prices of the portfolio and prices of portfolios depend on prices of all the assets in the portfolio, the aggregate demand curve for any asset is a function of the entire vector of N prices.

For arbitrary, non-clearing price vectors, the quantity vector $\mathbf{q} = D(\boldsymbol{\pi})$ may have both positive and negative components. We do not enforce a constraint that prices be nonnegative. Negative prices arise naturally in commodity markets, such as electricity, with limited storage and costly curtailment.

3.2 Illustrations

When there is only one asset or when all orders are for the same asset or portfolio, calculating an equilibrium price is easy. Since each order is a piecewise-linear function that demands a zero quantity at a sufficiently high or low price, the aggregate demand curve is continuous, weakly downward sloping, non-negative for sufficiently low prices, and non-positive for sufficiently high prices. Hence, the intermediate value theorem implies that there exists a clearing price, which may be either a single point or a closed interval. This price can be approximated arbitrarily closely by bisection or trial-and-error.

Figure 2 illustrates the simplicity of calculating clearing prices when all orders are for the same asset. Figure 2(a) shows a piecewise-linear demand curve and a piecewise-linear supply curve which intersect at a unique clearing price $p = \$45.00$ per share. The market trading volume is $x = 2.00$ shares per second. Figure 2(b) shows the combined demand and supply curve as one downward-sloping net demand curve, which has the same clearing price at a net quantity of zero. While the clearing price is easily calculated from the net demand curve, trading volume cannot generally be inferred from the net demand curve.

With many orders for different portfolios of multiple assets, calculating a clearing price is mathematically more complicated because the intermediate value theorem and the bisection

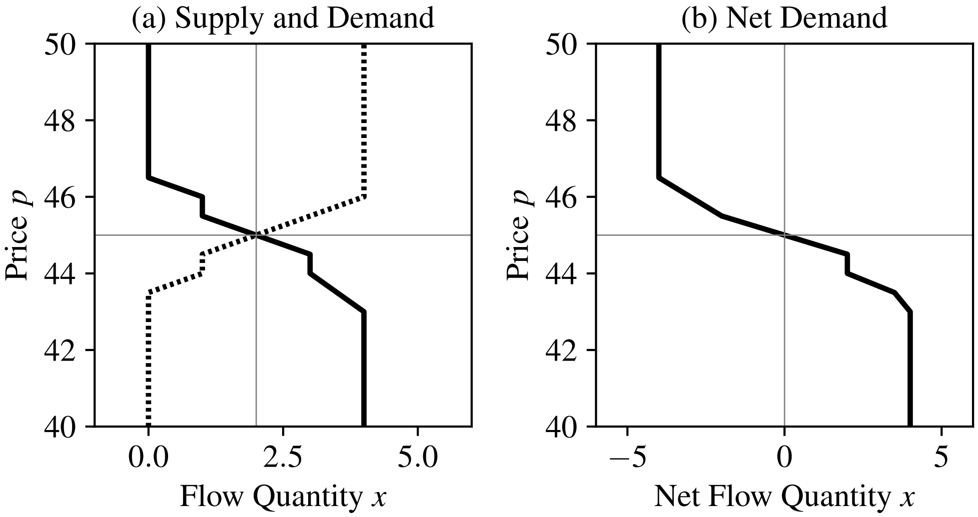

Figure 2: Plots of (a) a supply curve (dashed) intersecting a demand curve (solid) and (b) the net demand curve obtained by summing supply and demand. The supply and demand curves are each generated by summing together three orders. The clearing price of \$45.00 per share generates trading volume of 2.00 shares per second. The clearing price and quantity can easily be calculated by bisection or trial and error.

algorithm do not generalize in a natural way.

Example: Two Assets Consider an example with two assets. Suppose there are only two orders in the market, a buy order for asset A and sell order for asset B. The only clearing prices involve no trade because neither order can find another order to trade with. The clearing prices which support no trade are not unique because any high enough prices for asset A and low enough prices for asset B will imply that both orders are non-executable. Next, add to this example a pairs-trade order for a portfolio with negative weight on asset A and positive weight on asset B, where upper and lower limit prices are set such the only clearing prices involve no trade. The set of no trade prices is now the interior of the triangle defined by the three dotted lines in Figure 3. As indicated in grey, this region is the intersection of the three orders' halfplanes of no execution.

If the limits p_i^H and p_i^L on the pairs trade order order are changed, it is possible for trade to occur, as in Figure 4. Figure 4(a) shows the order for asset A as a solid line and plots the derived demand for asset A as a heavy dashed line. The derived demand takes into account the derived demand from the pairs-trade order by fixing the price of asset B at its market-clearing value. Figure 4(b) shows the order for asset B as a solid line and the derived demand, holding the price of asset A constant at its market-clearing level, as a heavy dashed line. The pairs trade order makes the assets substitutes. Therefore, when the price of asset B is fixed at a level above

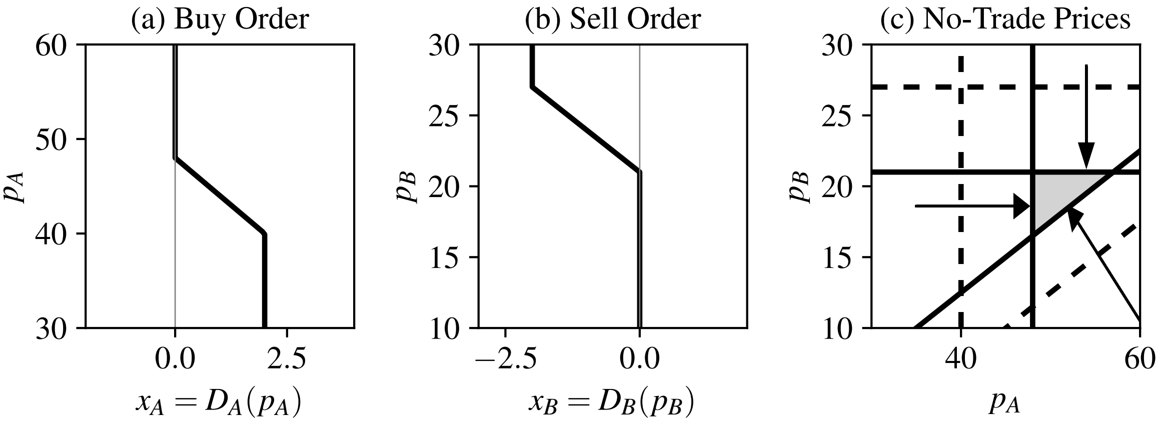

Figure 3: Plots of (a) a buy order for asset A; (b) a sell order for asset B; and (c) price pairs for the buy order, the sell order, and a pairs-trade order which sells A and buys B. In (c), the areas between the parallel solid and dashed lines indicate regions of partial execution of the orders. The solid lines bound the region of no execution and the dotted lines bound the region of full execution. The three orders generate no trade at any prices on the interior of the triangle defined by the intersection of the dashed lines. The arrows, which are proportional to the vectors of portfolio weights, indicate the direction in which the orders tend to push prices. In the Euclidian norm, the arrows are orthogonal to the solid lines they point to. As plotted, the arrows may not appear to be perpendicular to the lines they point to because one dollar on the vertical axis represents a different length than one dollar on the horizontal axis.

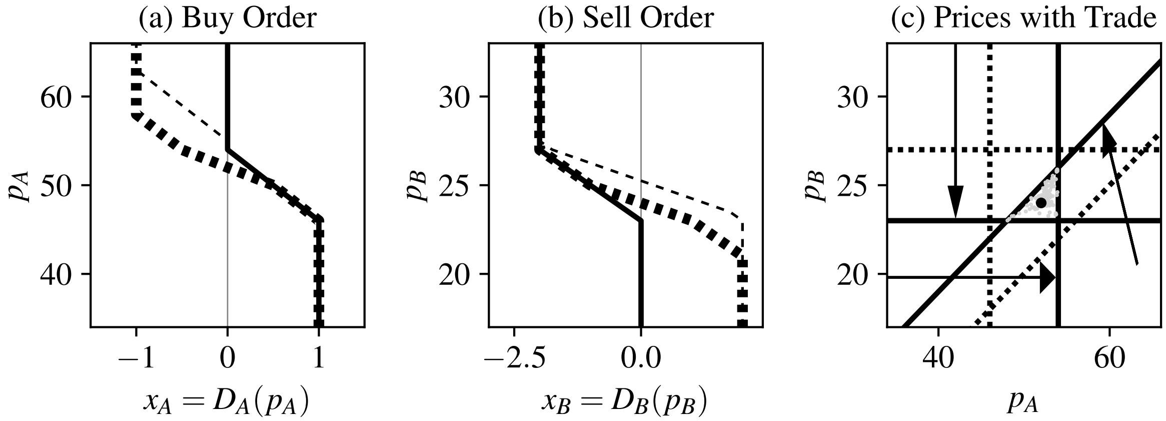

Figure 4: Plots of (a) a buy order for asset A, and sell order for asset B, and (c) price pairs for both assets. In (a) and (b), the solid line represents the order itself, the heavy dashed line represents the demand schedule generated by the order for the asset and the pairs-trade order fixing the price of the other asset at its market-clearing level, and the light dash lines represents the demand for asset A from the order itself and the pairs-trade order fixing prices for the other asset above the market clearing level. In (c), the solid lines indicate boundaries between no execution and partial execution of the buy order, the sell order, and the pairs-trade order. The dashed lines indicate boundaries between partial execution and full execution of the orders. Trade occurs at a price on the interior of the triangle defined by the intersection of the three solid lines. Arrows proportional to orders' portfolio weights indicate directions in which the order push prices.

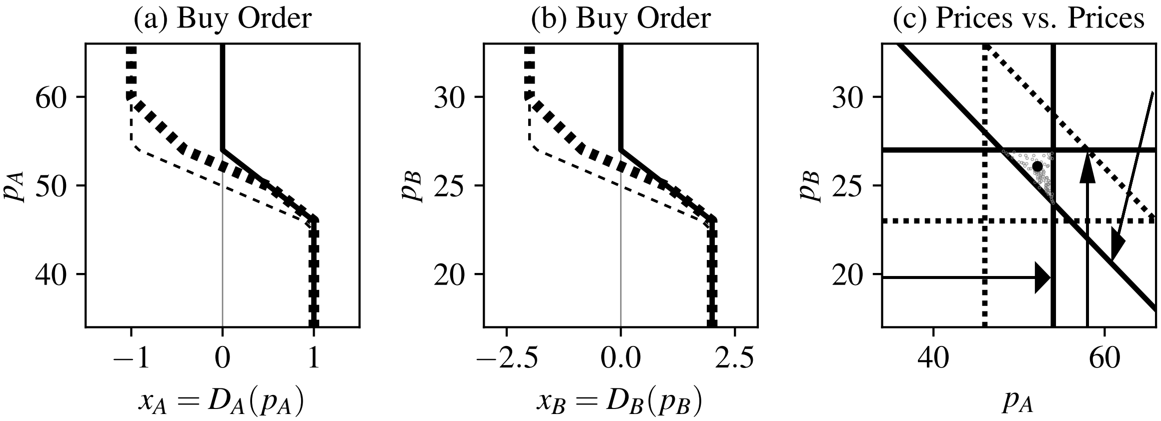

Figure 5: Plots of (a) a buy order for asset A and buy order for asset B, and (c) an index order to sell both assets. In (a) and (b), the solid line represents the order itself, the heavy dashed line represents the demand schedule generated by the order for the asset and the index order fixing the price of the other asset at its market-clearing level, and the light dash lines represents the demand for the asset from the order itself and the pairs-trade order fixing prices above the market clearing level for the other asset. In (c), the solid lines indicate boundaries between no execution and partial execution of the buy order, the sell order, and the index order. The dashed lines indicate boundaries between partial execution and full execution of the orders. Trade occurs at a price on the interior of the triangle defined by the intersection of the three solid lines. Arrows proportional to orders' portfolio weights indicate directions in which the orders push prices.

its equilibrium price, the derived demand for asset A is shifted towards higher prices as well. The same is true for asset B. This is depicted in Figures 4(a) and 4(b) as the thin dashed lines, which for both assets have prices which are the same or higher than the thick dashed line. In Figure 4(c), the region where all three orders are partially or fully executable is the interior of the triangle defined by the intersection of the solid lines defining the boundary between no execution and partial execution of the orders.

Now, consider the case when there are buy orders for each asset A and B and a sell order for the index portfolio of assets A and B. Let the price limits on the index order be such that trade occurs only in the region where all three orders are partially executable. The portfolio order makes the assets complements. Therefore, when the price of asset B is fixed at a level above its equilibrium price, the derived demand for asset A is shifted towards lower prices. The same is true for asset B. This is depicted in Figures 5(a) and 5(b) as the thin dashed lines, which for both assets have prices which are the same or lower than the thick dashed line. The parallel lines defining the infinite strip in which the index order is partially executable are downward sloping rather than upward sloping because keeping the portfolio price constant as asset prices change requires increasing the price of one asset and decreasing the price of the other asset.

The examples with two assets in Figures 3, 4, 5 illustrate principles which generalize to markets with more than two assets. The parallel lines defining regions of partial execution become parallel hyperplanes. The arrows, which point in directions proportional to portfolio weights defined by the orders, become vectors perpendicular to the hyperplanes.

The regions where prices are not unique, such as regions of no trade in Figures 3, are convex sets which may either be bounded (compact) or extend to infinity in one or more directions. It is this property of the set of clearing prices that enables simple tie-breaking rules. For example, nearest-to-the-prior-price-vector identifies a unique price vector within the set of clearing prices, since the set of clearing prices is convex.

3.3 Existence of Market Clearing Prices and Quantities

To show the existence of clearing prices, which then determine market clearing quantities, we formulate an optimization problem by imputing to each order "as-bid" preferences which define the dollar utility value of the number of portfolio units bought, then sum the utility functions across orders to obtain the objective function to be maximized.

In the range of prices where an order is partially executable, the demand is a linear function of prices. Therefore, a quadratic quasilinear utility function defines preferences. The constraints preventing overfilling or underfilling the order are linear inequality constraints. The constraint that markets clear are linear equality constraints. Putting this together mathematically results in the problem of maximizing a quadratic utility function subject to linear constraints. The quadratic utility function is the sum across orders of a one-variable quadratic utility function for each order. There are two inequality constraints for each order, one to prevent overfilling and the other to prevent negative portfolio quantities. There is one equality constraint for each asset.

Quadratic programs have been thoroughly studied and are well-understood. Given the structure of our problem, it is well known that unique utility maximizing quantities exist, and the solution implies Lagrange multipliers which correspond to clearing prices. A solution to the dual problem of calculating optimal (market-clearing) prices also exists and implies the same solution as the original ("primal") problem.

In the rest of this section, we describe these well-known theoretical results by mapping them into economic language which makes connections with economic concepts such as maximizing utility, consumer surplus, and market clearing.

Imputing utility functions to orders is a convenient mathematical modeling device. We proceed as though orders directly represent consumer preferences, even though, in practice, traders submit orders strategically. Thus, our methodology does not measure actual economic welfare and does not generate welfare results on market efficiency. Still, the method provides a practical approach to prove that clearing prices and quantities exist.

Pseudo-Utility Let $V_i(x)$ denote the dollar utility of order i from a trade rate of x in portfolio units per second. To find $V_i(x)$, we first define the marginal utility function $M_i(x)$ as the inverse demand curve, $p_i = M_i(x_i)$, where recall the order i demand curve is denoted by $D_i(p_i) = x_i$. In words, the inverse demand curve maps order i's trade rate $x \in [0, q_i]$ into prices $p \in [p_i^L, p_i^H]$.[6] Rearranging equation (2) we have:

$$M_i(x) := p_i^H - \frac{p_i^H - p_i^L}{q_i} x \qquad \text{for } x \in [0, q_i]. \tag{5}$$

The value of $M_i(x)$ measures marginal as-bid flow value in dollars per portfolio unit. Utility $V_i(x)$, as a function of the trade rate x, is defined as the integral of the marginal utility function for trade rate over the interval $[0, x]$:

$$V_i(x) := \int_0^x M_i(u)\, \mathrm{d}u \tag{6}$$

Since the marginal value is linear in x, the total value is quadratic and therefore strictly concave in x:

$$V_i(x) = p_i^H x - \frac{p_i^H - p_i^L}{2q_i} x^2 \tag{7}$$

We will think of $V_i(x)$ as defined for all $x \in \mathbb{R}$, with order specifications imposing the constraint $x \in [0, q_i]$.[7]

Value Maximization Our problem of finding clearing prices is formulated as two optimization problems, a primal problem of finding quantities which maximize “as-bid dollar value” and a dual problem of finding prices which minimize the cost of non-clearing prices. The first-order conditions for optimality of either of these two problems imply market clearing quantities and prices.

The exchange, acting analogously to a social planner in general equilibrium theory, chooses a vector of execution rates for all orders $\mathbf{x} = (x_1, \ldots, x_I)$ to maximize aggregate as-bid value, defined as the sum of pseudo-utility functions across orders,

$$V(\mathbf{x}) := \sum_{i=1}^{I} V_i(x_i) \qquad \text{for } \mathbf{x} \in \mathbb{R}^I, \tag{8}$$

[6] For trade rates in the interval $(0, q_i)$, the fact that the order chooses an interior quantity tells us that the order's as-bid marginal utility is equal to the corresponding price in the interval (p_i^L, p_i^H). The same logic extends to the boundary points 0 and q_i, corresponding respectively to prices p_i^H and p_i^L, by assuming as-bid utility is continuous.

[7] We could equivalently think of the domain of $V_i(x)$ as $x \in [0, q_i]$ or define $V_i(x) = -\infty$ for $x \notin [0, q_i]$.

subject to choosing quantities consistent with market clearing constraints and order execution rate constraints:

$$\max_{\mathbf{x}} V(\mathbf{x}) \qquad \text{subject to} \begin{cases} \sum_{i=0}^{I} x_i \mathbf{w}_i = \mathbf{0} & \text{(market clearing)} \\ x_i \in [0, q_i] \text{ for all } i & \text{(order execution rate),} \end{cases} \tag{9}$$

The objective function $V(\mathbf{x})$ is concave because it is a sum of concave functions.

Indeed, since the objective function is quadratic and the constraints are linear, this is a quadratic program (QP). To make this quadratic structure apparent using matrix and vector notation, let $\mathbf{W}$ denote the $N \times I$ matrix whose ith column is $\mathbf{w}_i$. Let $\mathbf{p}^H$ denote the column vector whose ith element is p_i^H. Let $\mathbf{D}$ denote the $I \times I$ positive definite diagonal matrix whose ith diagonal element is $(p_i^H - p_i^L)/q_i$. Then problem (9) may be written compactly as

$$\max_{\mathbf{x}} \left[\mathbf{x}^\top \mathbf{p}^H - \tfrac{1}{2}\mathbf{x}^\top \mathbf{D}\mathbf{x}\right] \qquad \text{subject to} \qquad \mathbf{W}\mathbf{x} = \mathbf{0} \qquad \text{and} \qquad \mathbf{0} \le \mathbf{x} \le \mathbf{q}. \tag{10}$$

We first show that quantities which maximize aggregate utility exist. Then we show that clearing prices exist by examining the dual problem to the utility maximization problem.

Theorem 1 (Existence and Uniqueness of Optimal Quantities)**.** *There exists a unique quantity vector* $\mathbf{x}^*$ *which solves the maximization problem* (10).

Proof. The problem has the following properties:

1. Compactness and convexity: The inequality constraints on trade rates define the Cartesian product of I intervals, $[0, q_1] \times \cdots \times [0, q_I]$, which is compact and convex. The market clearing conditions are linear constraints, which defines the intersection of hyperplanes. The intersection of a compact, convex set with hyperplanes is compact and convex. Thus, the set of vectors of trade rates $\mathbf{x}$ that satisfies all constraints is compact and convex.

2. Feasibility: No trade ($\mathbf{x} = \mathbf{0}$) generates well-defined utility for each order ($V_i(0) = 0$), clears markets and is allowed on each order. In this sense, no-trade is feasible.

3. Strict concavity: Each function $V_i(x_i)$ is quadratic and therefore strictly concave for all $x_i \in \mathbb{R}$. Since V is the sum of V_i across i, the function V is concave on the domain $\mathbb{R}^I$ and thus also on the compact and convex subset defined by the constraints.

It is a well-known principle of convex analysis (Boyd and Vandenberghe (2004); Bertsekas (2009); Nocedal and Wright (2006)) that a strictly concave objective function on a non-empty compact and convex set has a unique maximizing vector $\mathbf{x}^*$. □

Our approach makes the problem compact by assuming that traders are not interested in trading additional quantities beyond some very favorable level of prices. This is like putting upper and lower bounds on quantities and linear combinations of quantities.

To prove that clearing prices exist, we exploit the duality between the problems of finding optimal quantities and prices. For this, we define a Lagrangian function of the vector of quantities $\mathbf{x}$ with three constraints: (1) the market clears ($\sum_{i=1}^{I} x_i \cdot \mathbf{w}_i = \mathbf{0}$); (2) the order execution rate is greater than or equal to zero ($\mathbf{x} \geq \mathbf{0}$); (3) the order execution rate is less than or equal to the maximum ($\mathbf{x} \leq \mathbf{q}$). In vector notation, the Lagrangian is defined by

$$L(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := \mathbf{x}^{\top}\mathbf{p}^{H} - \tfrac{1}{2}\mathbf{x}^{\top}\mathbf{D}\,\mathbf{x} - \boldsymbol{\pi}^{\top}\mathbf{W}\mathbf{x} + \boldsymbol{\mu}^{\top}\mathbf{x} + \boldsymbol{\lambda}^{\top}(\mathbf{q} - \mathbf{x}). \tag{11}$$

Since the multipliers associated with the market clearing equality constraint have the economic interpretation as market prices for assets, we use the notation $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ for these multipliers. Two vectors of order-execution-rate multipliers, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_I)$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_I)$, are associated with inequality constraints on order execution rates, with two constraints for each order.

The dual problem associated with the primal problem of maximizing aggregate utility (10), is then defined by

$$\hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := \max_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \text{for} \qquad \boldsymbol{\pi} \in \mathbb{R}^N, \qquad \boldsymbol{\mu} \geq \mathbf{0}, \qquad \boldsymbol{\lambda} \geq \mathbf{0}. \tag{12}$$

The dual problem is a minimization problem with infimum g^* defined by

$$g^* := \inf_{\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}} \hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \text{subject to} \qquad \boldsymbol{\pi} \in \mathbb{R}^N, \qquad \boldsymbol{\mu} \geq \mathbf{0}, \qquad \boldsymbol{\lambda} \geq \mathbf{0}. \tag{13}$$

The dual problem (13) is formulated as an infimum rather than minimum because we have not yet shown that there exists a solution ($\boldsymbol{\pi}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^*$) which attains the infimum.

Theorem 2 (Existence of clearing prices)**.** *There exists at least one optimal solution* $(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *to the dual problem* (13)*. The solutions* $\mathbf{x}^*$ *and* $(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *are a primal-dual pair which satisfies the strict duality relationship*

$$g^* = V(\mathbf{x}^*). \tag{14}$$

Proof of Theorem 2. The primal problem has the following properties:

1. Concavity: The objective function $V(\mathbf{x})$ is strictly concave.

2. Finite solution: The primal objective is the sum of a finite number of concave quadratic functions. Since each quadratic function is bounded above, the solution to the primal problem is bounded above.

3. Linear constraints: The minimum execution rate constraint $\mathbf{x} \geq \mathbf{0}$, the maximum execution rate constraint $\mathbf{x} \leq \mathbf{q}$, and the market clearing constraint $\mathbf{W}\mathbf{x} = \mathbf{0}$ are all linear.

4. Feasibility: No trade ($\mathbf{x} = \mathbf{0}$) is feasible because it clears the markets and is allowed on

each order.[8]

It is a standard result from convex programming that a concave primal problem, a finite supremum on the primal problem, feasibility, and linear constraints guarantee that a solution to the dual problem exists and has the same optimal value as the supremum to the primal problem even if a solution to the primal problem does not exist like it does in our problem; see Boyd and Vandenberghe (2004), Bertsekas (2009, Proposition 5.3.4, p. 173), Nocedal and Wright (2006, Theorem 16.4, p. 464). Since Theorem 1 guarantees that a solution to the primal problem does exist, the solution to the primal problem has the same value as the solution to the dual problem. □

There are three Lagrange multipliers in this problem: $\boldsymbol{\pi}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$. The multiplier on the market clearing condition $\boldsymbol{\pi}$ is the vector of prices for all assets. The other multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ ensure that orders are not underfilled ($\mathbf{x} < \mathbf{0}$) or overfilled ($\mathbf{x} > \mathbf{q}$).

It follows from the market-clearing equality constraint that the solution $\boldsymbol{\pi}$ to the Lagrangian is a set of clearing prices. Another way to see this is that if the market were not to clear, $\boldsymbol{\pi}$ can be tweaked slightly to decrease $\hat{G}$.[9] Since the constraints on the order are inequality constraints, each of the individual multipliers must be nonnegative ($\mu_i \geq 0$ and $\lambda_i \geq 0$ for all i).

Theorem 2 does not guarantee that clearing prices are unique. The set of clearing prices is convex and may be unbounded, as in Figure 3 above. A trivial example occurs when all orders are buy orders for individual assets, and there are no sell orders. Then any sufficiently high price clears the market with zero trade. There may also be cases where the clearing price is not unique even when trade occurs. A trivial example occurs when there is one fully executable buy order and one fully executable sell order for the same asset, with the same quantities q and the lower limit price on the buy order strictly above the upper limit price on the sell order. We discuss a tie-breaking rule to pick a unique price in the next section.

The duality between the primal problem and the dual problem has the intuition of a zero-sum game played by the price-setting Walrasian auctioneer and hypothetical traders who optimize execution of linear orders ($x \in \mathbb{R}$) without upper and lower limits ($x \in [0, q_i]$) but instead with linear incentives provided by the multipliers λ_i and μ_i. To interpret the dual problem economically, think of the exchange as a Walrasian auctioneer who seeks to quote prices $\boldsymbol{\pi}$ which clear markets. If we think of the auctioneer as attempting to solve the dual problem, the auctioneer also quotes multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ which are designed to prevent overfilling or underfilling orders. We can think of the dual objective as the cost to the auctioneer of quoting non-clearing prices over and above the cost of clearing prices. This cost is defined by imagining that the auc-

[8] Feasibility does not require a strict interior point (Slater's condition) because the constraints are linear in this problem (linear constraint qualification).

[9] Note that prices can be positive, negative, or zero.

tioneer takes into its own inventories any net uncleared quantities, then, once clearing prices are known, liquidating these quantities at a loss by walking up or down the orders' marginal valuation curves, buying or selling at a loss. Similarly, the Lagrange multipliers are set to prevent overfilling or underfilling the order. Since the Lagrangian pretends that each order's value function $V_i(x)$ is quadratic for all $\mathbb{R}$, not just the interval $[0, q_i]$, the exchange reduces its costs by making the multipliers larger when the order is being overfilled or underfilled, relative to the multipliers which solve the dual optimization.

4 Equilibrium Characterization and Computation

In this section we address the following issue: We have existence of market-clearing prices, as well as uniqueness of these prices up to a convex set, from results in Section 3. Can we compute these prices quickly?

There are many economic settings where market-clearing prices are known to exist but where it is also known they can be hard to find quickly (Scarf and Hansen (1973)). More modern work in computer science has focused on the complexity of computing Brouwer and Kakutani fixed points (Daskalakis, Goldberg, and Papadimitriou (2009); Budish, Cachon, Kessler, and Othman (2017)) and supports the claim that computing competitive equilibrium prices can be computationally difficult.

There are also, of course, many economic settings where market-clearing prices are known to exist and trivial to compute. For example, in a single-asset allocation environment with continuous downward sloping demand and continuous upward sloping supply, computing the market clearing price is trivial. Formally, you can use the bisection method.

In our setting, we have downward sloping demand schedules for portfolios, which guarantees that quantities are unique and prices are unique up to a convex set. Under minor regularity conditions satisfied by our problem, this implies that the gradient method, which is equivalent to Walrasian tatonnement, is guaranteed to converge, unlike in the traditional general equilibrium environment studied by Scarf and Hansen (1973). In this sense, our problem is immediately seen to be "easier" than the traditional general equilibrium environment and is more analogous to the simple supply-and-demand environment for one asset. In our problem, however, there may be hundreds of asset prices and tens of thousands of orders. The gradient method performs poorly in such high-dimensional problems.

Although our proposal does not contemplate the exchange trading as a market maker, it is useful to add the exchange as a market maker for theoretical and computational reasons. Exchange trading solves the tiebreaker problem by helping select a price when the set of market clearing prices in nonempty and perhaps unbounded. Exchange trading can speed up the gradient method. The speed-up, however, is not enough to solve our problem in less than one second. Exchange trading also makes it possible to use interior point methods. Interior point methods are well-known in the convex optimization literature to be fast and efficient (Nesterov (2004); Bertsekas (2009); Boyd and Vandenberghe (2004); Gondzio (2012)).

The plan of this section is as follows. Subsection 4.1 shows that the gradient method is equivalent to applying Walrasian tatonnement to a variation of the dual problem; it converges to a solution is not fast enough for our application. Subsection 4.2 motivates interior point methods with the exchange as market maker and describes how the method works. Subsection 4.3 discusses the results of our simulations. It turns out that prices can be computed in less than one second for hundreds of assets and tens of thousands of orders, with a very small amount of exchange trading needed to help stabilize the algorithm numerically.

4.1 Walrasian tatonnement, the gradient method, and the dual problem

Economists often approach the problem of finding market clearing prices by using the familiar process of Walrasian tatonnement: Starting with an initial price guess, proceed iteratively by revising the price guess in the direction of calculated net excess demands until uncleared quantities are close enough to zero.

Two questions about tatonnement naturally come to mind: (1) Do market clearing prices exist? (2) Does a tatonnement process converge to these prices efficiently enough to be useful as a practical algorithm? Intuition suggests that tatonnement should work correctly because demand schedules are downward sloping for every portfolio.

These questions can be answered formally by mapping the problem of finding market clearing prices into a convex optimization problem for which the first-order conditions equate net excess demands to zero. The relevant convex optimization problem is the problem of maximizing consumer surplus.

The remainder of this subsection spells out details for the following analysis. Define a "gains-from-trade" function $G(\boldsymbol{\pi})$ which maps prices $\boldsymbol{\pi}$ into aggregate consumer surplus. The derivative of this gains function is the negative of net excess demands, $\nabla G(\boldsymbol{\pi}) = -D(\boldsymbol{\pi})$. This makes Walrasian tatonnement equivalent to the gradient method for optimizing convex functions since the gradient method adjusts prices in the direction of net excess demand, just like tatonnement. Since the the gains function is closely related to the dual problem, Theorem 2 already shows that clearing prices exist. Textbook theorems from convex optimization describe how the efficiency of the gradient method depends on the smoothness of the derivative of the objective function. Since demands are piecewise linear, the gains functions has a continuous derivative which satisfies a Lipschitz condition. This is enough to guarantee a much faster rate of conver-

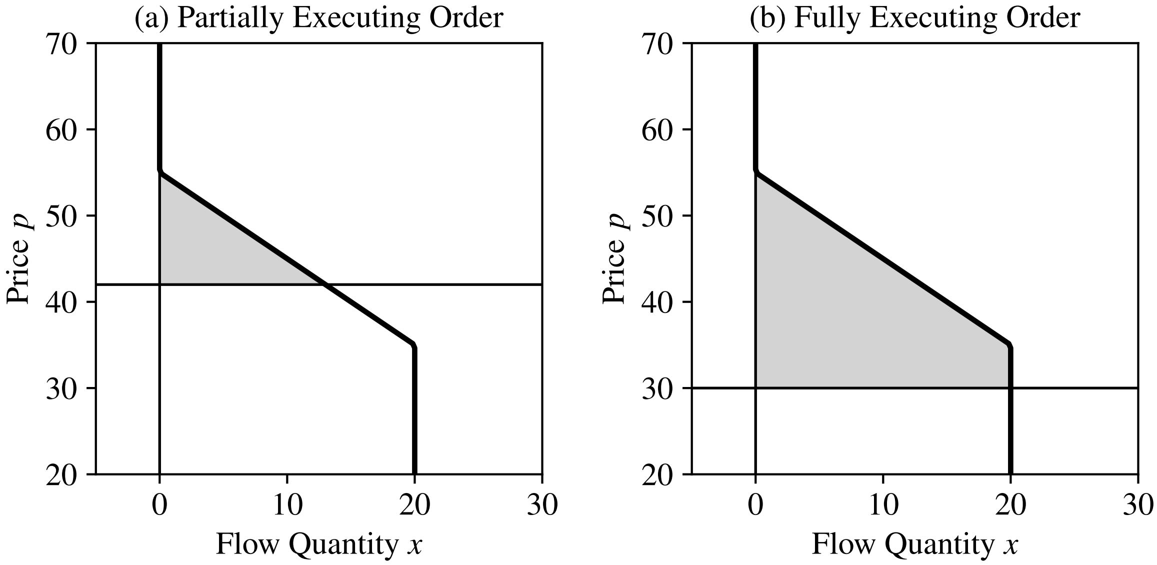

Figure 6: Plots of gains from trade for (a) a partially executing order and (b) and fully executing order. The gains from trade are represented by the area of the regions shaded in grey.

gence than for the difficult general-equilibrium theory problems discussed by Scarf and Hansen (1973). Unfortunately, this derivative is not smooth enough to guarantee a rate of convergence fast enough for our problem, which requires calculating clearing prices in less than one second.

Gains function For each order, define a "gains function" which measures the consumer surplus from optimally trading a portfolio at price p as the integral of the difference between price and marginal utility, or equivalently as the value function minus the cost of buying the quantity executed:

$$G_i(p) := \int_0^{D_i(p)} \big(M_i(z) - p\big)\,\mathrm{d}z. \tag{15}$$

$$:= V_i\big(D_i(p)\big) - p \cdot D_i(p). \tag{16}$$

To provide intuition, Figure 6 illustrates the gains-from-trade for a buy order. The dollar gain is the area of a shaded region, which is a trapezoid when the the price is so low that the order is fully executing ($p_i^H \leq p$), a triangle when the order is partially executing ($p_i^L < p < p_i^H$), and exactly zero when the price is so high that the order is not executing ($p_i^H \leq p$).

The explicit solution for $G_i(p)$ is

$$G_i(p) := \begin{cases} q_i \left(\frac{1}{2}(p_i^H - p_i^L) - p\right) & \text{for } p < p_i^L \qquad \text{(fully executable)}, \\ \dfrac{q_i\,(p_i^H - p)^2}{2(p_i^H - p_i^L)} & \text{for } p_i^L \le p \le p_i^H \qquad \text{(partially executable)}, \\ 0 & \text{for } p_i^H \le p \qquad \text{(non-executable)}. \end{cases} \tag{17}$$

The three cases in equation (17) match up exactly with the three cases for the multipliers, where $\mu_i > 0$ corresponds to no execution, $\lambda_i > 0$ corresponds to full execution, and $\mu_i = \lambda_i = 0$ corresponds to partial execution.

Define the aggregate gains-from-trade function $G(\boldsymbol{\pi})$ as the sum of the individual gains-from-trade functions with the portfolio price calculated from the portfolio weights and price vector, $p_i = \boldsymbol{\pi}^\top \mathbf{w}_i$:

$$G(\boldsymbol{\pi}) = \sum_{i=1}^{I} G_i\left(\boldsymbol{\pi}^\top \mathbf{w}_i\right). \tag{18}$$

By construction, the gradient of the gains function is minus the aggregate demand function,

$$\nabla G(\boldsymbol{\pi}) = -D(\boldsymbol{\pi}), \tag{19}$$

which is defined everywhere and is piecewise linear. Therefore, the exchange can try to find market clearing prices by solving the set of N equations $D(\boldsymbol{\pi}) = \mathbf{0}$. like a Walrasian auctioneer.

The gains function and the dual problem We can answer the question whether $\nabla G(\boldsymbol{\pi}) = \mathbf{0}$ has a solution by relating the gains function to the dual problem. The gains function is obtained from the dual problem by "maximizing out" the multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$, which builds the inequality constraints on order execution $\mathbf{0} \le \mathbf{x} \le \mathbf{q}$ into the structure of the gains function itself. Theorem 2 already establishes that a price vector minimizing the gains function does indeed exist:

Theorem 3 (Gains function)**.** *The gains function is related to the dual problem by*

$$G(\boldsymbol{\pi}) = \min_{\boldsymbol{\lambda},\boldsymbol{\mu}} \hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \textit{subject to} \qquad \boldsymbol{\lambda} \ge \mathbf{0}, \qquad \boldsymbol{\mu} \ge \mathbf{0}. \tag{20}$$

Every clearing price vector $\boldsymbol{\pi}^*$ *satisfies*

$$\boldsymbol{\pi}^* = \operatorname*{argmin}_{\boldsymbol{\pi} \in \mathbb{R}^N} G(\boldsymbol{\pi}). \tag{21}$$

The set of clearing prices is a nonempty, closed convex set which may be unbounded.

Proof. See Appendix A □

The second derivative of the gains function $\nabla^2 G(\boldsymbol{\pi})$, defined everywhere except at the kinks in the piecewise linear demand curves, is an $N \times N$ matrix given by

$$\nabla^2 G(\boldsymbol{\pi}) = -\nabla D(\boldsymbol{\pi}) = -\sum_{i=1}^{I} \frac{\mathrm{d}D_i(p_i)}{\mathrm{d}p_i} \cdot \mathbf{w}_i \mathbf{w}_i{}^\top. \tag{22}$$

The objective function $G(\boldsymbol{\pi})$ is convex in $\boldsymbol{\pi}$. This is consistent with $\nabla^2 G(\boldsymbol{\pi})$ being the positively-weighted sum of $N \times N$ rank-one positive semi-define matrices $\mathbf{w}_i \mathbf{w}_i{}^\top$ and therefore itself positive semidefinite. At points where G is not differentiable, the gains function is also convex (and has nonunique subgradients).

The exchange's problem of calculating market clearing price is equivalent to minimizing, not maximizing, the gains function. This may seem counterintuitive. If the exchange is maximizing aggregate utility, what is the economic intuition for its minimizing and not maximizing the gains function? The answer comes from using standard duality intuition to consider the costs incurred by the exchange when it trades as a market maker at non-clearing prices.

When the exchange quotes market-clearing prices, the allocation of quantities across traders is Pareto optimal according to the competitive equilibrium imputed to their imputed utility functions and demand schedules. The exchange itself does not trade at all. It functions purely as an agent for calculating prices.

Now imagine that, after honoring all trades at market-clearing prices, the exchange changes the prices, and all traders adjust their quantities. Every trader adjusts quantities to achieve weakly better consumer surplus by buying low and selling high according to their downward-sloping demand schedules for portfolios. The exchange pays for this increased utility by buying high and selling low. This implies that the exchange loses more money relative to market clearing prices than the traders gain in consumer surplus. Intuitively, for a small price change, we can think of a trader who changes quantities as gaining a small triangle of surplus while the exchange incurs a small rectangle of costs twice as large as the triangle. In this way, by minimizing the gains function $G(\boldsymbol{\pi})$, the exchange maximizes aggregate utility when its own losses are taken into account.[10]

Walrasian Tatonnement and the Gradient Method Now we address the question whether prices can be efficiently calculated using tatonnement. Under tatonnement, a Walrasian auctioneer announces tentative prices, traders in aggregate respond with their quantities, the auctioneer adjusts prices in a direction proportional to net excess demand, and the process continues until convergence to equilibrium prices occurs. This description of Walrasian tatonnement

[10] The argument presented here applies to portfolios a logic similar to the argument of Friedman (1960) defending destabilizing speculation.

is identical to the gradient method of optimization when the gradient method is used to minimize the gains function. The gradient method proceeds iteratively by adjusting prices in the direction of the negative gradient. Walrasian tatonnement applied to the gains function corresponds exactly to the gradient method because the gradient is the negative of excess demands.

Standard textbook results on optimization show that the gradient method is not expected to work well on our problem. Small enough price adjustments in the direction of net excess demands do result in an improvement in the objective function. Unfortunately, our objective function is not smooth enough to guarantee rapid convergence. Since the gains function has piecwise-linear derivative, it is continuously differentiable, and the derivative satisfies a Lipschitz condition $|\nabla G(\boldsymbol{\pi}+\Delta\boldsymbol{\pi}) - \nabla G(\boldsymbol{\pi})| < L|\Delta\boldsymbol{\pi}|$ for some Lipschitz constant L. Now let $\boldsymbol{\pi}_k$, $k = 0, 1, \ldots$, be a sequence of gradient-method iterations starting with intital guess $\boldsymbol{\pi}_0$. Nesterov (2004, Corollary 2.1.2, p. 70) proves a representative theorem providing a pessimistic guarantee concerning how well the gradient method works under these assumptions:

Theorem 4. *Let G be a convex with continuously differentiable gradient satisfying a Lipschitz condition with constant L. Using step size $1/L$, the the error after k iterations of the gradient method $G(\boldsymbol{\pi}_k) - G(\boldsymbol{\pi}^*)$ is related to the error of the initial guess $\boldsymbol{\pi}_0 - \boldsymbol{\pi}^*$ by*

$$G(\boldsymbol{\pi}_k) - G(\boldsymbol{\pi}^*) \leq \frac{2L\|\boldsymbol{\pi}_0 - \boldsymbol{\pi}^*\|^2}{k+4}. \tag{23}$$

This theorem says that halving the error $G(\boldsymbol{\pi}_k) - G(\boldsymbol{\pi}^*)$ may require approximately doubling the number of iterations. Thus, reducing the error by a factor of one million may require approximately one million iterations, a prohibitively large number. Of course, this theorem is providing a worst-case guarantee. As discussed in Appendix B.1, (Nesterov, 2004, Chapter 2) proves other theorems, which also give pessimistic results.

While different variations of the gradient method (or tatonnement) depend on the formula for calculating a step size for each iteration, the implications of convergence theorems are also pessimistic for other step size methodologies.

4.2 Interior-Point Methods with the Exchange as Market Maker

Our flow trading proposal does not contemplate the exchange trading as a market maker like a traditional NYSE specialist. Instead, we think of the exchange as an agent who calculates prices for all the other market participants but does not trade on its own account.

Nevertheless, it turns out to be practically necessary to add the exchange as a market maker for computational reasons associated with numerical efficiency and stability. In this subsection, we show why adding the exchange as market maker is useful from perspectives which are both practical and theoretical. In the next section, we present simulation results which show that only a tiny amount of exchange trading is needed.

As a practical matter, a tiny amount of exchange trading can solve the "tiebreaker problem" of choosing market prices when the set of market prices is not unique and possibly unbounded. If the exchange has a very small partially executable order active for every asset at every relevant price vector, then the exchange's demands will guarantee unique prices for all assets. This solves the tiebreaker problem in a nice way because the exchange can trade in a manner which reduces the difference between the current market clearing price and, say, last period's market clearing price, choosing "reasonable" prices when the choices picked by an algorithm could otherwise seem arbitrary and create unwanted transitory volatility in clearing prices from auction to auction.

A deeper theoretical reason for exchange trading is that it makes algorithms for calculating market clearing prices more efficient. Theoretically, the gradient method can be sped up by using information about the curvature of the objective function, obtained from the hessian, to improve the search direction. Newton's method, when applied to an objective with continuously differentiable second derivative satisfying a Lipschitz condition, converges quadratically if the current guess is close enough to the optimum. Quadratic convergence means that the number of significant digits doubles on each iteration.

Unfortunately, our problem does not have a continuous hessian. Furthermore, if we consider a difficult problem in which the difference between upper and lower limit prices on all orders, $\mathbf{p}^H - \mathbf{p}^L$, is very small, the hessian will be exactly zero at almost all price vectors and therefore provide no useful information about how to improve the gradient search direction. Nevertheless, it can be shown that adding exchange trading, by making the objective function more convex, can improve the convergence rate of the gradient method by intuitively guaranteeing better search directions. For details, see Appendix B.1, which discusses the relationship between exchange trading and the concept of "strict convexity." While exchange trading does improve the rate of convergence of the gradient method, it does not make it fast enough to solve for clearing prices in less than one second. Therefore something better is needed.

Much faster convergence can potentially be obtained by using interior point methods of optimization. Interior point methods replace the inequality constraints in the primal problem with smooth penalty functions added to the objective. In our problem, we delete the constraints $\mathbf{x} \geq \mathbf{0}$ and add a penalty function of the form $\bar{v} \cdot \log(\mathbf{x})^\top \mathbf{1}$ to the objective. Since the domain of the function $\log(x)$ is $x > 0$, not $x \geq 0$, the penalty function is a "log-barrier function" which prevents the inequality constraints from ever holding with equality. Similarly, the penalty function $\bar{v} \cdot \log(\mathbf{q} - \mathbf{x})$ replaces the constraint $\mathbf{x} \leq \mathbf{q}$ and enforces the strict inequality $\mathbf{x} < \mathbf{q}$.

Interior point methods are based on the following intuition: In the limit $\bar{v} \to 0$, the solution to the altered problem converges to the solution to the original problem, with $\mathbf{x} \to \mathbf{0}$ for non-executable orders and $\mathbf{x} \to \mathbf{q}$ for fully executable orders. Furthermore, even though the original problem was not smooth enough for Newton's method, the new problem is smooth enough. Even better, by shifting attention from the dual problem of finding optimal prices to the primal problem of finding optimal quantities, the interior point method looks deep into the order book to use information about the whether an order is "likely" to be partially executable and therefore relevant for price discovery at the margin. It does this by attaching a low weight to orders where $\mathbf{x}_k$ is close to $\mathbf{0}$ or close to $\mathbf{q}$.

Interior point methods require that there exists a feasible allocation on the interior of the constraint set; such an allocation satisfies market clearing and also satisfies inequality constraints strictly, $\mathbf{0} < \mathbf{x} < \mathbf{q}$. Our natural initial guess is no-trade ($\mathbf{x} = \mathbf{0}$). No-trade is feasible since it satisfies market clearing and the inequality constraints, but it does not lie on the interior of the inequality constraint set because $\mathbf{x} = \mathbf{0}$ is on the boundary. Our proof of Theorem 2 used a linear constraint qualification. If we knew that the order book had a market clearing allocation on the interior of the inequality constraint set, we could have used Slater's condition instead. Slater's condition requires a non-empty interior assumption like that used in general equilibrium theory and like that used to guarantee that interior point methods work. Unfortunately, Slater's condition does not hold for some obvious and likely cases, such as one or more orders to buy some illiquid asset but no orders willing to sell it.

A simple way to deal with this issue is to add the exchange as a market maker with linear demand and supply for each asset. Formally, the exchange's demand function for asset n can be expressed as a linear function of the price

$$y_n = \epsilon_n(\pi_{0n} - \pi_n), \tag{24}$$

where ϵ_n is a small positive number defining the slope of the exchange's demand schedule as a function of the price, π_{0n} is a price below which the exchange buys and above which it sells, and y_n is the quantity traded by the exchange (positive for buying, negative for selling). If ϵ_n is small, the exchange does not trade much. The value of ϵ_n can vary across assets if the exchange provides a different level of liquidity to different assets.

In matrix notation, the exchange's demands for all assets can be written $\mathbf{y} = \boldsymbol{\epsilon}(\boldsymbol{\pi}_0 - \boldsymbol{\pi})$, where $\boldsymbol{\epsilon}$ is the positive definite matrix whose diagonal is $(\epsilon_1, \ldots, \epsilon_N)$. The exchange's demand function can be justified by imputing to the exchange a quadratic utility function $\mathbf{y}^\top \boldsymbol{\pi}_0 - \frac{1}{2}\mathbf{y}^\top \boldsymbol{\epsilon}^{-1}\mathbf{y}$, adding this utility to the primal objective function, adding the quantity traded $\mathbf{y}$ to the market clearing condition, and leaving the inequality constraints unchanged (since the small quantities potentially bought or sold are theoretically unbounded). The new primal problem is

$$\max_{\mathbf{x},\mathbf{y}} \left[\mathbf{x}^{\mathsf{T}}\mathbf{p}^{H} - \tfrac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{D}\mathbf{x} + \mathbf{y}^{\mathsf{T}}\boldsymbol{\pi}_0 - \tfrac{1}{2}\mathbf{y}^{\mathsf{T}}\boldsymbol{\epsilon}^{-1}\mathbf{y}\right] \qquad \text{subject to} \qquad \mathbf{W}\mathbf{x} + \mathbf{y} = \mathbf{0}, \qquad \mathbf{0} \le \mathbf{x} \le \mathbf{q}. \tag{25}$$

Adding the exchange as market maker makes it possible for non-clearing quantities to be in the feasible set since the exchange can trade the uncleared quantities. For example, any allocation $\mathbf{x} = \alpha \cdot \mathbf{q}$ is allowed when $0 < \alpha < 1$, even though it is not a feasible interior point without the exchange clearing the market.

Our interior point method changes our original primal problem in three ways: (1) It adds the exchange's utility function to the objective function; (2) it adds the exchange's quantities traded to the market-clearing condition, and (3) it replaces the inequality constraints with log-barrier penalty functions. The modified maximization problem is the modified primal problem (25) with inequality constraints replaced by log-barrier penalty functions:

$$\max_{\mathbf{x},\mathbf{y}} \left[\mathbf{x}^{\mathsf{T}}\mathbf{p}^{H} - \tfrac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{D}\mathbf{x} + \mathbf{y}^{\mathsf{T}}\boldsymbol{\pi}_0 - \tfrac{1}{2}\mathbf{y}^{\mathsf{T}}\boldsymbol{\epsilon}^{-1}\mathbf{y} + \bar{\nu}\cdot\log(\mathbf{x})^{\mathsf{T}}\mathbf{1} + \bar{\nu}\cdot\log(\mathbf{q}-\mathbf{x})^{\mathsf{T}}\mathbf{1}\right] \qquad \text{subject to} \qquad \mathbf{W}\mathbf{x}+\mathbf{y} = \mathbf{0}. \tag{26}$$

Note that the exchange's preferences are defined over assets while the customer orders define preferences over portfolios. The log-barrier function $\bar{\nu}\cdot\log(\mathbf{x})^{\mathsf{T}}\mathbf{1}$ replaces the inequality constraint $\mathbf{0} \le \mathbf{x}$, and the log-barrier function $\bar{\nu}\cdot\log(\mathbf{q}-\mathbf{x})^{\mathsf{T}}\mathbf{1}$ replaces the inequality constraint $\mathbf{x} \le \mathbf{q}$. In the limit as $\bar{\nu} \to 0$, the solution to this problem is the solution to the modified primal problem of maximizing utility with the exchange added to the problem.

Solution Methodology and Karush–Kuhn–Tucker (KKT) conditions How does the approach used by the interior point method change the manner in which the solution is characterized, without changing the solution itself.

For our original problem, the following theorem shows that market clearing prices and quantities are characterized by the unique solution of the Karush–Kuhn–Tucker (KKT) conditions.

Definition 1. *The* Karush–Kuhn–Tucker (KKT) Conditions *for for primal feasibility, dual feasibility, primal optimality, and complementary slackness are*

$$\mathbf{W}\mathbf{x}^{*} = \mathbf{0}, \qquad \mathbf{0} \le \mathbf{x}^{*} \le \mathbf{q} \qquad \textit{(Primal Feasibility)}, \tag{27}$$

$$\boldsymbol{\pi}^{*} \in \mathbb{R}^{N}, \qquad \boldsymbol{\lambda}^{*} \ge \mathbf{0} \qquad \boldsymbol{\mu}^{*} \ge \mathbf{0}, \qquad \textit{(Dual Feasibility)} \tag{28}$$

$$\mathbf{p}^{H} - \mathbf{D}\mathbf{x}^{*} - \mathbf{W}^{\mathsf{T}}\boldsymbol{\pi}^{*} + \boldsymbol{\mu}^{*} - \boldsymbol{\lambda}^{*} = \mathbf{0} \qquad \textit{(Primal Optimality)} \tag{29}$$

$$\boldsymbol{\lambda}^{*}\cdot(\mathbf{q}-\mathbf{x}^{*}) = \mathbf{0}, \qquad \boldsymbol{\mu}^{*}\cdot\mathbf{x}^{*} = \mathbf{0} \qquad \textit{(Complementary Slackness)}. \tag{30}$$

Theorem 5 (Karush–Kuhn–Tucker (KKT) Conditions)**.** *Any solution of the KKT conditions* (27)–(30) *for quantities* $\mathbf{x}^* := (x_1^*, \ldots, x_I^*)$ *and multipliers* $(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *is a solution to both the primal problem and dual problem. There exists at least one solution to the KKT conditions. The solution for optimal quantities* $\mathbf{x}^*$ *is unique.*

Proof of Theorem 5. This is a straightforward consequence of Theorems 1 and 2, which imply that a unique optimal primal solution $\mathbf{x}^*$ exists and some optimal dual solution $(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ exists, and these solutions form a primal dual pair with the same optimized value; see Bertsekas (2009, Theorem 5.34(b), p. 173) .

Equation (29) presents first-order conditions for the primal optimality problem

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^I}{\operatorname{argmax}}\ L(\mathbf{x}, \boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*), \tag{31}$$

where L is defined in equation (11). Since the problem is quadratic and strictly concave, the first-order conditions are necessary, and they are sufficient if the problem has a solution. The sufficiency follows from the existence of clearing prices and quantities. □

In a nutshell, Theorem 5 suggests that (1) solving the primal problem for quantities $\mathbf{x}$, (2) solving the dual problem for prices $\boldsymbol{\pi}$, and (3) solving the KKT equations are three different approaches to solving the same problem. Indeed, it is straightforward to use the KKT equations to obtain the solution to the primal problem from a solution to the dual problem, or vice versa.

For the modified problem solved by the interior point method, the solution is characterized by the same KKT conditions (27)–(30) modified in three ways: (1) Add the exchange's demand to the first-order condition. (2) Add the exchange's quantities traded to the market-clearing condition. (3) Replace the complementary slackness condition $\boldsymbol{\mu}^* \cdot \mathbf{x}^* = \mathbf{0}$ with $\boldsymbol{\mu}^* \cdot \mathbf{x}^* = \bar{v} \cdot \mathbf{1}$, and then let $\bar{v} \to 0$. The modified KKT conditions are

$$\mathbf{W}\mathbf{x}^* + \mathbf{y}^* = \mathbf{0}, \qquad \mathbf{0} \le \mathbf{x}^* \le \mathbf{q}, \qquad \mathbf{y}^* \in \mathbb{R}^N \qquad \text{(Primal Feasibility)}, \tag{32}$$

$$\boldsymbol{\pi}^* \in \mathbb{R}^N, \qquad \boldsymbol{\lambda}^* > \mathbf{0} \qquad \boldsymbol{\mu}^* > \mathbf{0}, \qquad \text{(Dual Feasibility)} \tag{33}$$

$$\mathbf{p}^H - \mathbf{D}\mathbf{x}^* - \boldsymbol{\epsilon}^{-1}\mathbf{y}^* - \mathbf{W}^\top \boldsymbol{\pi}^* + \boldsymbol{\mu}^* - \boldsymbol{\lambda}^* = \mathbf{0} \qquad \text{(Primal Optimality)} \tag{34}$$

$$\boldsymbol{\lambda}^* \cdot (\mathbf{q} - \mathbf{x}^*) = \bar{v} \cdot \mathbf{1}, \qquad \boldsymbol{\mu}^* \cdot \mathbf{x}^* = \bar{v} \cdot \mathbf{1}, \qquad \bar{v} > 0, \qquad \bar{v} \to 0 \qquad \text{(Complementary Slackness)}. \tag{35}$$

Interior point methods proceed by solving the revised problem for finite $\bar{v} > 0$ while at the same time pushing $\bar{v}$ closer and closer to zero. At each iteration, the guess for $\mathbf{x}$ remains an interior point, with $x \to \mathbf{0}$ and $\mathbf{x} \to \mathbf{q}$ for nonexecutable and fully executatble orders, respectively.

It can be shown theoretically that interior point methods have favorable complexity (see

Nesterov (2004, Chapter 4); Bertsekas (2009), Boyd and Vandenberghe (2004)). In a survey of interior point methods, Gondzio (2012) compares worst-case complexity results for interior point methods with gradient methods. For interior point methods, the maximum number of iterations has an upper bound proportional to $\sqrt{I}\log(1/\epsilon)$, where ϵ is the proportion by which the error is reduced (Theorem 3.1). In practice, the dependence on I is $\log(I)$, not $\sqrt{I}$. For gradient methods, the corresponding worst-case complexity result is $O(1/\epsilon)$ or $O(1/\epsilon^2)$. In practice, consistent with this result, gradient methods have difficulty achieving high accuracy.

Our simulations use our own straightforward Python implementation of the interior point methodology in the CVXOPT package, as described by Vandenberghe (2010). Both the Python programming language and the CVXOPT package are free and publicly available. Our implementation is tailored to our specific quadratic program, which has an invertible diagonal matrix $\mathbf{D}$ and simple "Euclidean cone" constraints $\mathbf{0} \leq \mathbf{x} \leq \mathbf{q}$.

In brief, the algorithmic strategy is to linearize the KKT conditions (which has $3I + N$ equations), solve the linearized system[11] with $\bar{v} = 0$ to obtain a search direction which reduces $\bar{v}$, then take a step $\Delta\mathbf{x}$ which keeps the best guess an interior point and maintains the constraint $\bar{v} > 0$. Since the KKT conditions are essentially first-order conditions, the linearized approximation is a version of Newton's method. At each step, the multipliers are expressed as functions of $\mathbf{x}$, easy invertibility of the diagonal matrix $\mathbf{D}$ allows $\mathbf{x}$ to be expressed as a simple function of $\boldsymbol{\pi}$, and substituting the solution for $\mathbf{x}$ into the market clearing condition reduces the problem to solving an $N \times N$ positive definite system for a price update to $\boldsymbol{\pi}$ using a Cholesky decomposition. The positive-definite matrix to be decomposed changes with each iteration because it is constructed by implicitly assigning weights to each order based on values of multipliers. The weights are close to zero if the order is anticipated either to be fully executable or non-executable; the weights are closer to one if the order is anticipated to be partially executable. A new Cholesky decomposition is needed on each iteration to incorporate updated weights from the most recent iteration into calculation of the new search direction. This is how the algorithm's Newton method uses information about orders deep in the order book. For more discussion, see Appendix 4.2.

[11] The revised KKT system is nonlinear in the unknowns $\boldsymbol{\pi}$, $\mathbf{x}$, $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$, and $\bar{v}$ only because the revised complementary slackness condition involves element-by-element multiplication of $\mathbf{x}$ by $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$. We use the predictor-correction adjustment described by Mehrotra (1992) to approximate the effect of this nonlinearity in two steps using only one Cholesky decomposition.

4.3 Simulation Results

We have implemented a simulation framework to test the practical performance of flow trading. Calculations were performed on an ordinary office workstation.[12] The goal of the simulations is to show that a base-case problem with 500 assets and 30,000 orders can reliably be solved in less than one second. We regard meeting this goal as "proof of concept" that flow trading is computationally feasible.

The simulation framework has two components. First, we simulate an order book in which one-third of the orders are for individual assets, one-third pairs trades, and one-third for index portfolios. Second, we use the interior point method, with modest exchange trading, to obtain clearing prices and quantities. To measure computation time, we only count the time to run the quadratic programming algorithm, excluding the time to construct the order book, under the assumption that a realistic production environment will run the optimizer with dedicated computer resources.

We have tried to make the optimization problem difficult by having huge variation in trader-provided liquidity across assets, including assets with no orders; poorly conditioned matrices resulting from index orders, with even worse conditioning due to mixing equally- and value-weighted indexes; equally-weighted- and value-weighted industry indexes which create poor conditioning along different dimensions from volume-based portfolios; small differences between p_i^H and p_i^L, which make the gradient of the gains function look like a difficult-to-optimize step function; and very little exchange trading to stabilize the problem. Details of the simulation methodology are provided in Appendix B.3.

Computation Results Figure 7 presents plots of computation time for 500 simulated orders books in each of two panels.

In the first panel, the number of assets varies from 10 to 10,000, with the number of orders held constant at 30,000. Execution time is about 0.30 seconds for 500 assets, approaches one second for about 2000 assets, and approaches 10 seconds for about 10,000 assets. As the number of assets exceeds 1000, the plot becomes linear in logs with a slope of 2 or more. The slopes is related to the fact that the algorithm requires a dense Cholesky decomposition of an $N \times N$ matrix every iteration. The Cholesky decomposition is an $O(N^3)$ algorithm, and calculating the $N \times N$ matrix to be decomposed is itself also burdensome.

In the second panel, the number of orders varies from 100 to 1.5 million, with the number of assets held constant at 500. Execution time is less 0.20 seconds if there are fewer than about 10,000 orders, approaches one second for about 100,000 orders, and is about 10 seconds for 1

[12] The workstation has an AMD Ryzen Threadripper 3960X processor, 24 cores running at 3.8GHz, and 128GB of memory running at 3600MHz.

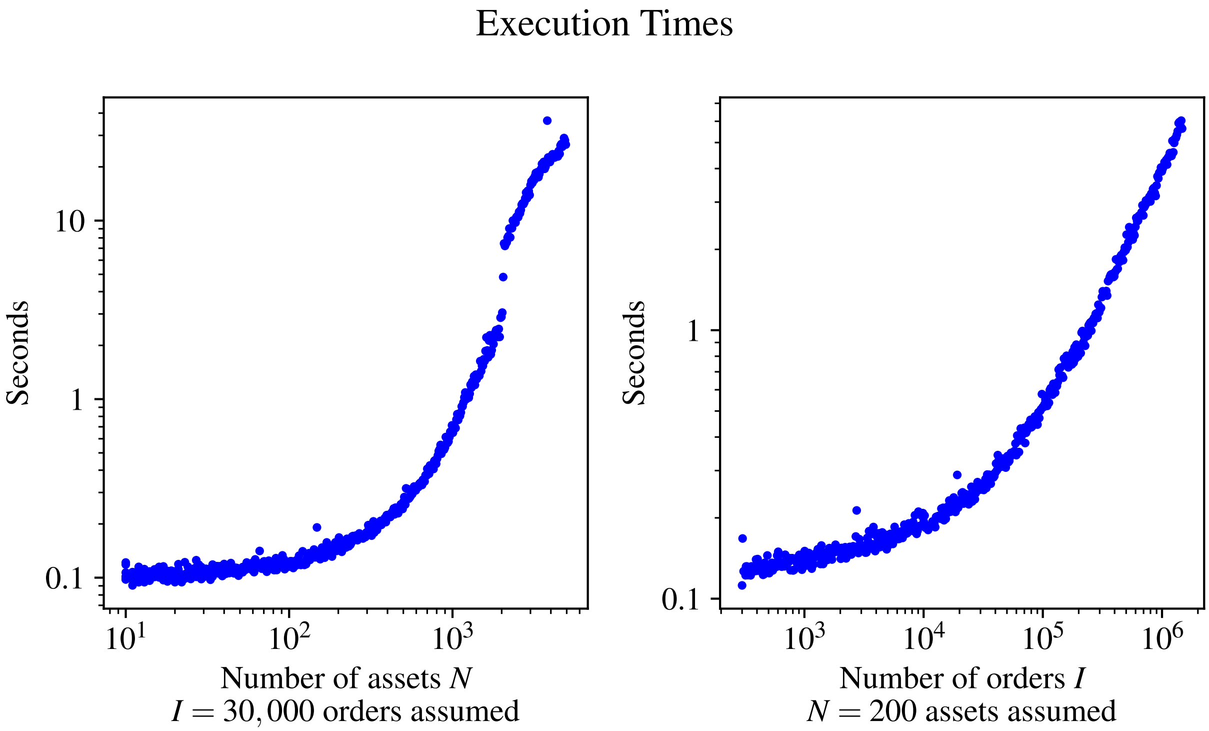

Figure 7: Execution time for simulated market outcomes.

million orders. As the number of orders increases, execution time is approximately linear in logs with a slope of about one: Computation time is approximately proportional to the number of orders when the number of orders is large.

At the far left of both plots, where the problem is as small as allowed, the computation time is about 0.10 seconds. A significant fraction of this is overhead from the Python interpreter, which could be reduced by using an efficient compiled language like C++.

In these simulations, the exchange's share of dollar volume as market maker is small. The exchange's share is less than 10^{-7}, often much less. Since the exchange's initial price guesses $\boldsymbol{\pi}_0$ are not very accurate indicators of clearing prices $\boldsymbol{\pi}^*$, exchange trading in a realistic environment is likely to be an even smaller share of volume.

Uncleared quantities, due to convergence tolerance settings or numerical error, are another form of exchange trading since the exchange is assumed to execute all orders at calculated prices. These uncleared quantities are typically an order of magnitude smaller than the quantities traded by the exchange as market maker. In a few cases, they are about the same size. The modest exchange trading in these simulations helps to stabilize numerical calculations and speeds up calculations slightly by reducing the number of iterations. If settings are changed so that the exchange trades significantly less, the Cholesky decomposition occasionally fails because a matrix which is positive definite in theory may not be positive definite numerically. The number of iterations also increases because the search directions are less efficient, presumably due to numerical error. Exchange trading stabilizes the numerical calculations by reducing the condition number on the matrix to which the Cholesky decomposition is applied. Intuitively, this matrix measures the depth of the market in all portfolio directions. Adding modest exchange trading presumably improves the condition number on this matrix by adding some liquidity to assets or portfolios which other orders are not expressing an interest to trade at prices relevant for the current iteration.[13]

Preliminary results suggest that finding clearing prices takes modestly fewer iterations when the difference between $\mathbf{p}^H$ and $\mathbf{p}^L$ is larger. It also takes modestly fewer iterations when exchange trading becomes more and more economically significant. In general, the number of iterations required increase somewhat in I and N but is insensitive to other assumptions.

Except for the exchange's role in solving the tiebreaker problem of picking a price when it is not unique, it is not the purpose of our simulations to find prices that are economically reasonable. Instead, our simulations are designed to stress the algorithm by posing difficult problems, then see how long it takes the algorithm to find market clearing prices, even if they are not economically reasonable. Indeed, as our stress simulations are designed to bring about, many of our prices are "unreasonable" in the sense that some are thousands of percent higher or lower (implying negative prices) than the expected midpoint of bids and offers. This occurs because some illiquid assets have few or no orders, with their prices defined by how they appear in portfolio orders. In practice, like the way current exchanges operate, we expect market participants to place orders which prevent unreasonable prices or excessive intraday price volatility, not the market clearing algorithm or the exchange itself trading as a price-stabilizing market maker.

We interpret these results as being a proof of concept that flow trading is computationally practical when the market clears at intervals of one second. In a production environment in the future, faster CPU speeds, better parallelized sparse matrix operations, and more refined algorithms should make it easier to calculate clearing prices with even greater speed.[14]

5 Portfolio Orders in the CARA-Normal Framework

Our portfolio flow orders requires that the demand for a given portfolio depends only on the price of the portfolio. This may appear restrictive given that the demand for portfolios can generally depend on all prices of the assets. In this section, we show that despite the restriction portfolio orders can be used to implement optimal portfolios.

[13] The handful of outliers in Figure 7 could be eliminated by having the exchange trade more aggressively. We do not yet understand the apparent discontinuity which occurs in the first panel when there are about 3000 assets.

[14] Even though our algorithm is set up to use multiple processors, it does not seem to do so efficiently, for reasons we do not yet fully understand but may be related to difficulties parallelizing sparse matrix operations.

5.1 The Static CARA-Normal Framework

In a canonical CARA-normal framework, widely used in the economics and finance literature (e.g., Grossman and Stiglitz (1980)), we study the implications of trading portfolios. For simplicity we focus on a static setting, in which there is no distinction between trading in quantities and trading in flows, and interpret it as a single batch. The literature on dynamic strategic trading (Vayanos (1999); Du and Zhu (2017); Kyle, Obizhaeva, and Wang (2018)) shows that trading gradually over time is optimal.

Consider an individual trader placing orders in the market. She has constant absolute risk aversion preferences with risk aversion A and zero initial wealth.[15] There are N risky assets and one safe asset, whose return is normalized to one. Let $\mathbf{v}$ denote the vector of risky assets' payoffs. The trader has subjective beliefs that $\mathbf{v}$ is jointly normally distributed with mean $\mathbf{m}$ and variance-covariance matrix $\mathbf{\Sigma}$.

The trader chooses her optimal portfolio to maximize her expected utility, given by

$$\max_{\boldsymbol{\omega}} \mathrm{E}\left[-\exp^{-A(\mathbf{v}-\boldsymbol{\pi})^{\top}\boldsymbol{\omega}}\right], \tag{36}$$

where $\boldsymbol{\pi}$ is the vector of asset prices. We will begin by supposing that traders know the realized prices $\boldsymbol{\pi}$ and choose the optimal quantity demanded for those prices. Later, traders can implement the demand by submitting the whole demand schedule that specifies quantities contingent on realized prices.

The joint normality assumption allows us to transform above into the quadratic optimization problem:

$$\max_{\boldsymbol{\omega}} \left[(\mathbf{m}-\boldsymbol{\pi})^{\top}\boldsymbol{\omega}-\frac{1}{2}A\,\boldsymbol{\omega}^{\top}\mathbf{\Sigma}\boldsymbol{\omega}\right]. \tag{37}$$

Assume, for now, that the trader is a perfect competitor, taking the market clearing prices as given.[16] Then the first order condition implies that the optimal portfolio is given by

$$\boldsymbol{\omega}^{*}=(A\,\mathbf{\Sigma})^{-1}(\mathbf{m}-\boldsymbol{\pi}). \tag{38}$$

Notice that the optimal portfolio determines the demand for each asset as a linear function that depends on the prices of all assets.

Orders for Individual Assets Since the demand for each asset depends on all prices, if traders are restricted to using orders for individual assets that are a function only of the individual as-

[15] This is without loss of generality since there is no wealth effect in CARA preferences.

[16] Below we show that the main results do not change when traders behave strategically, taking into account their price impact.

set's price as in the market design of current stock exchanges, they cannot trade optimally according to equation (38). Recall, in a single asset setting, the ability to make price-contingent orders allows traders to choose the optimal trade as if they observe the realized price. In a multi-asset setting, implementing the optimal trade requires making orders contingent on the prices of all assets. Thus, if restricted to use asset orders contingent only on the asset's price, traders must bear the risk of obtaining quantities that are far from the optimal demand. See also the discussion in Section 1.1 for the related literature that studies the equilibrium implications of such restrictions in asset orders.

Rotation Now suppose traders can submit orders for portfolios that are a function of the price of the portfolio as in our proposal. Can they then achieve the optimal trade described in equation (38)? Below we show that the answer is yes. For this, we need to "rotate" the asset space such that it is spanned by independent portfolios.

Since the variance-covariance matrix $\mathbf{\Sigma}$ is positive semidefinite, its singular value decomposition has a form

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Delta}\mathbf{U}^{\top}, \tag{39}$$

where $\mathbf{U}$ is an orthonormal matrix, and $\mathbf{\Delta}$ is a diagonal matrix with nonnegative elements. Let $K \le N$ denote the rank of $\mathbf{\Sigma}$, let δ_i denote the ith nonzero diagonal entry of $\mathbf{\Delta}$, and let $\mathbf{u}_i$ denote the corresponding column of $\mathbf{U}$.[17] Then we have

$$\mathbf{\Sigma}^{-1} = \sum_{i=1}^{K} \frac{1}{\delta_i} \mathbf{u}_i {\mathbf{u}_i}^{\top}. \tag{40}$$

Using this, we can express the optimal portfolio in equation (38) as

$$\boldsymbol{\omega}^* = \sum_{i=1}^{K} \left(\frac{{\mathbf{u}_i}^{\top}\mathbf{m} - {\mathbf{u}_i}^{\top}\boldsymbol{\pi}}{A\,\delta_i} \right) \mathbf{u}_i, \tag{41}$$

which is a combination of demand schedules for portfolios. Here, $\mathbf{u}_1, \ldots, \mathbf{u}_K$ are portfolios of the assets, which themselves are the "rotated" assets. Since they are independent of one another (and there is no wealth effect in CARA preferences), the optimal portfolio chooses the demand for each of them separately as if in a single-asset model. That is, the optimal demand for the ith portfolio is given by

$$\frac{1}{A\,\delta_i} ({\mathbf{u}_i}^{\top}\mathbf{m} - {\mathbf{u}_i}^{\top}\boldsymbol{\pi}), \tag{42}$$

where δ_i, ${\mathbf{u}_i}^{\top}\mathbf{m}$, and ${\mathbf{u}_i}^{\top}\boldsymbol{\pi}$ correspond to the variance, the expected payoff, and the price of the

[17] When K is strictly less than N (i.e., the matrix $\mathbf{\Sigma}$ is positive semidefinite but not positive definite), we can use the pseudo-inverse instead of the inverse to define the demand function.

portfolio $\mathbf{u}_i$, respectively. Since the demand for each portfolio only depends on the portfolio's price, traders can achieve the optimal trade in equation (38) by utilizing K orders for portfolios where each order is a function of that portfolio's price. Recall, in our proposed market design, we require portfolio orders to be downward sloping. Since the optimal demand for each portfolio in equation (41) is decreasing in the portfolio's price, the demand is indeed downward sloping.

The theorem below summarizes the results.

Theorem 6. *Consider a static CARA-normal framework in which a trader believes that the variance-covariance matrix of the asset payoffs has rank K. Then the trader's optimal portfolio (equation* (38)*) can be represented as the sum of K downward-sloping demand schedules for portfolios, each of which depends only on that portfolio's price (equation* (41)*).*

Practical Implementation We can decompose the expected utility from the optimal portfolio into the contribution of each rotated asset. From substituting the optimal portfolio in equation (41) into equation (37), we can express the expected utility from trading at prices $\boldsymbol{\pi}$ as

$$\sum_{i=1}^{K} \frac{1}{2A} \left(\frac{\mathbf{u}_i{}^{\top}\mathbf{m} - \mathbf{u}_i{}^{\top}\boldsymbol{\pi}}{\sqrt{\delta_i}} \right)^2 . \tag{43}$$

This shows that the benefit of each portfolio is determined by its squared Sharpe ratio as perceived by the trader.[18] In practice, traders may select a few portfolios, which they perceive to have a sufficiently high Sharpe ratio (more precisely, its absolute value), and choose to trade only those portfolios rather than all of the K portfolios.

Strategic Trading Thus far, we have assumed that traders are perfect competitors, behaving as if they have no price impact. In practice, trades can indeed move prices, and many institutional traders dedicate considerable time and resources to measure and mitigate their price impact. Now we show that portfolio orders can still be used to implement the optimal portfolio when traders behave strategically, taking into account their price impact.

Following the literature (for example, Kyle (1989); Malamud and Rostek (2017)), we assume that traders believe that their price impact is linear in the quantity they trade. We further assume that the matrix of price impact is positive semidefinite.[19] That is, for each trader, there is

[18] Recall, the Sharpe ratio refers to the risk premium (i.e., the expected return minus risk free rate) divided by the standard deviation. Here, the risk free rate is zero since the safe asset's return is normalized to one.

[19] Malamud and Rostek (2017) show that when the variance-covariance matrix is the same for all traders, each trader's equilibrium price impact matrix is proportional to the variance-covariance matrix, which implies that all price impact matrices are positive semidefinite. It is left for future study to determine under what conditions the price impact matrix is positive semidefinite in a more general setting.

an $N \times N$ positive semidefinite matrix $\mathbf{\Lambda}$, such that

$$\boldsymbol{\pi} = \boldsymbol{\pi}_0 + \mathbf{\Lambda}\, \boldsymbol{\omega}, \tag{44}$$

where $\boldsymbol{\pi}_0$ is the vector of hypothetical prices that would prevail if the trader were not to trade, and the nth row of $\mathbf{\Lambda}$ corresponds to the marginal impact of trading assets 1 to N on the price of asset n. With a slight abuse of notation, we use the demand schedule $\boldsymbol{\omega}$ to also refer to the actual quantities that a trader trades at given prices.

With price impact, the trader's optimal strategy is a slight modification of the competitive solution in equation (38), given by

$$\boldsymbol{\omega}^* = (A\, \mathbf{\Sigma} + \mathbf{\Lambda})^{-1}(\mathbf{m} - \boldsymbol{\pi}). \tag{45}$$

Since the sum of two positive semidefinite matrices is also positive semidefinite, $A\, \mathbf{\Sigma} + \mathbf{\Lambda}$ is positive semidefinite. Thus, we can use singular value decomposition to rotate the asset space such that it is spanned by independent portfolios. Then the same logic as above implies that the optimal portfolio can be implemented by combining portfolio orders that only depend on the portfolio's price. The number of required portfolio orders corresponds to the rank of $A\, \mathbf{\Sigma} + \mathbf{\Lambda}$.

Theorem 7. *Consider a static CARA-normal framework in which a trader believes that her price impact is linear and positive semidefinite (equation* (44)*). Then the strategic trader's optimal portfolio (equation* (45)*) can be represented as the sum of downward-sloping demand schedules for portfolios, each of which depends only on that portfolio's price.*

Recall, when proving the existence and uniqueness of market clearing quantities in Section 3, we treat the orders as if they represent the traders' true valuations. This, as mentioned earlier, is just a solution technique and does not necessarily imply that we can infer the traders' true valuations from their orders. Strategic trading is one example that generates the gap between true and as-bid valuations.

5.2 Approximations for General Preferences and Limitations

The two key properties of CARA preferences we use in the arguments above are that they are strictly concave and that there are no wealth effects. Thus, our logic above extends to an arbitrary strictly concave twice continuously differentiable quasilinear utility function over assets. Since quasilinearity implies no wealth effect, the demand for each independently rotated asset can be found as if in a single asset model and thus only depends on the price of the rotated asset. With strictly concave utility, the demand for each asset can be locally approximated as a downward-sloping linear demand schedule. As market prices change, the local approximation also changes, in which case the trader will have to revise the portfolio orders.

However, portfolio orders will not be able to approximate the optimal portfolio of every concave utility function closely. In particular, with wealth effects, the demand for rotated assets may depend on the prices of the other assets and may also increase in their prices.

6 Discussion of Implementation and Policy Issues

Information Policy Information policy is typically discussed in terms of pre-trade transparency and post-trade transparency. Concerning post-trade transparency, we propose that the exchange publish the trading volume and clearing price of each asset promptly after the quantities and price have been calculated. In addition, the exchange may also publish information about the slope of the net demand curve for each asset, from which traders can make inferences about the price impact costs of their orders. The exchange does not publish information about the identity of traders. If clearing prices can be calculated in one-half second and prices published immediately, then traders would have another half-second to process this information to submit orders to trade at the next batch auction.

Pre-trade transparency in a market with batch auctions works differently from how it works with the traditional market. Traditional exchanges publish best bid and best ask prices and quantities. Such publication makes sense because an order may arrive at any time and execute against the published quotes. Published quotes are actionable for some positive duration. With frequent batch auctions, there is no trading between auctions. Therefore published quotes would not be disciplined by the possibility of incoming orders to trade at the quotes. Furthermore, calculating derived bid-ask spreads for assets from all portfolio orders, including orders for multiple assets, imposes a computational burden that cannot be met in real-time. Finally, since auctions occur frequently, the post-trade information about price and volume is much more relevant for deciding on orders in the next auction. Thus, pre-trade transparency for the auction at time $t+1$ consists of the post-auction information disseminated from the auction at time t.

With arbitrary portfolio orders, information about the depth of the order book is inherently complex because the depth of the order book for a portfolio cannot be inferred from the depth of the order book for individual assets. The exchange might publish limited depth information about each asset and also limited depth information about a fixed list of popular portfolios.

If the exchange does not publish much information about the depth of the order book, traders might measure the depth themselves by changing their orders for one second to see what happens. Such information has an opportunity cost which is lower when auctions are held more frequently.

Trust Flow trading has the desirable trust property that traders can infer from the history of their own orders and the history of prices the exact quantities they should have traded. By contrast, executable orders in current markets do not always execute when other orders execute at the same price. This erodes trust and market confidence, particularly among traders without state-of-the-art speed tools, whose orders are more apt to lack time priority and therefore get poorer execution.

Flow trading has a minor trust issue about whether messages sent a few milliseconds before the end of the batch interval are received in time to participate in that auction. Participants have no incentive to wait for the last milliseconds before placing the order. More importantly, with a short batch interval, the economic importance of any single auction is minor.

Fairness In traditional markets, the concept of "bid-ask spread" captures many of the features participants complain about as unfair. When there is a minimum tick size and the bid-ask spread is one-tick wide, buyers and sellers cannot offer price improvement by quoting better prices between the best bid price and best offer price. Instead, buyers and sellers queue up at the best bid and offer, where the fastest traders have the highest priority in the queue. Slow traders perceive this as frustrating and unfair. In dealer markets, dealers do not allow customers to post limit orders to trade directly with other customers. Instead, customers must trade with dealers in transactions where the dealer buys at the bid price and sells at the offer price. Customers complain that dealer markets are unfair because dealers have privileges that customers do not have. With flow trading, the concept of bid-ask spread is irrelevant when trade occurs because the market demand schedule for the asset is continuous and strictly downward sloping. All trades clear at the same price. All executable orders execute. Customers can increase the quantities they trade by offering small price improvements because there are typically additional quantities for purchase or sale at slightly improved prices. With flow trading, there still are trading costs. Trading faster requires offering better prices, which makes clearing prices move, which creates price impact.

offering better prices, which makes clearing prices move, which creates price impact.

Price Continuity as an Objective Traditional exchanges, such as the NYSE, have claimed price continuity as a market objective. Customers prefer price continuity precisely because they do not trust the integrity of order execution. If a customer saw a trade at a low price compared to recent prices, the customer would logically infer that the customer's own order was selling at the bad price and the NYSE specialist or another trader on the floor of the exchange was buying. The customer might also have inferred that "fast market" conditions were declared, which relieved his broker of the obligation to respect the limit price on his own resting order, which would have otherwise bought at the low price.

With flow trading, transitory price discontinuities benefit customers with orders that execute slowly over many batch auctions by allowing the orders to execute at better prices. For example, if prices for a particular asset are higher at one auction and lower at the next auction by the same price increment, resting executable customer limit orders trades the same combined quantity (by linearity) at the two auctions, but the average price of execution improves because a larger quantity is executed at the better price and a smaller quantity at the worse price.

Temporary price discontinuities can result from the arrival of overly urgent orders that have significant temporary price impact. Under a market design with flow trading, traders have strong incentives to place patient orders and to protect themselves from unfavorable prices by adjusting limit prices p_i^H and p_i^L to tolerable levels.

Regulatory Objectives The U.S. Securities and Exchange Commission (SEC), which regulates securities markets, pursues various general policy objectives, including economic efficiency, competition, capital formation, maintaining trust and confidence, and investor protection.

Flow trading is consistent with all of these objectives. It leads to economic efficiency by reducing wasteful expenditure on fast data feeds, communication technologies, and trading algorithms. It does this by decreasing the arms race among traders to pick off orders and by reducing the messages needed to implement dynamic trading strategies. It increases competition by providing customers, large and small, with a venue to trade small quantities at low cost. Flow trading is consistent with the current demand of small investors to trade fractions of shares and construct diversified portfolios consisting of tiny positions in many stocks. It makes capital formation more efficient by increasing market liquidity, which encourages markets to produce information about which firms can deploy capital most profitably. It promotes trust and confidence in markets by having all customers trade at the same transparent price. And it protects investors from poor order execution by making quality of order execution easy for customers to measure.

7 Conclusion

This paper has introduced a new market design for trading financial assets, such as stocks, bonds, futures, and currencies. It combines three elements: flow orders from Kyle and Lee (2017); frequent batch auctions from Budish, Cramton, and Shim (2015); and a novel language for trading portfolios of assets. Technical foundations for the proposed market design include existence and uniqueness results, computational results, and microfoundations for portfolio orders.

The combination of flow orders and frequent batch auctions yields a market design in which time is discrete and prices and quantities are continuous. The status quo market design has these reversed. As has been widely documented, treating time as a continuous variable and imposing discreteness on prices and quantities causes significant complexity, inefficiency, and rent-seeking in modern financial markets. Policy debates on the arms race for trading speed, the proliferation of complex order types, the importance of proprietary market data and exchange access, the cat-and-mouse game between institutional investors and high-frequency traders, and the internalization of retail investors' order flow, all relate to continuous time and discrete prices and quantities.

The novel language for portfolio orders is on the one hand rich enough to allow traders to directly express many important kinds of trading demands — customized ETFs, pairs trades, general long-short strategies, general market-making strategies, all with tunable urgency — while also allowing for guaranteed existence of equilibrium prices and quantities and their fast computation. This seems to us a useful new point on the frontier of language design, i.e., an attractive tradeoff between expressiveness and computability. Language design has been an active area of research and we hope there are further breakthroughs for financial-market applications in the future, possibly incorporating this paper's insights about what features of a portfolio language are important for existence and fast computability.

An open topic left for future research is the efficiency and welfare consequences of portfolio trading. We conjecture there are two main efficiency benefits. First, complexity and cost benefits of allowing market participants to directly express many common trading demands, which reduces systems complexity and the need for costly intermediation. Second, portfolio orders make it more efficient for sophisticated financial market participants to endogenously link prices and liquidity provision for correlated assets. Portfolio orders enable, for example, Bertrand competition on the cost of executing a Buy A, Sell B pairs trade, which is impossible under the status quo market design. We conjecture this will provide a liquidity and price discovery benefit for the market.

References

Arrow, Kenneth J., and Gerard Debreu. 1954. "Existence of an Equilibrium for a Competitive Economy." *Econometrica,* 1: 265–290.

Baldwin, Elizabeth, and Paul Klemperer. 2019. "Understanding preferences: 'Demand types', and the existence of equilibrium with indivisibilities." *Econometrica,* 87(3): 867–932.

Bertsekas, Dimitri P. 2009. *Convex Optimization Theory.* Athena Scientific Belmont.

Bichler, Martin. 2017. *Market Design: A Linear Programming Approach to Auctions and Matching.* Cambridge University Press.

Black, Fischer. 1971. "Toward a Fully Automated Exchange, Part I." *Financial Analysts Journal,* 27: 29–34.

Boyd, Stephen, and Lieven Vandenberghe. 2004. *Convex optimization.* Cambridge University Press.

Budish, Eric, and Judd B Kessler. forthcoming. "Bringing real market participants' real preferences into the lab: An experiment that changed the course allocation mechanism at Wharton." *Management Science.*

Budish, Eric, Gérard P Cachon, Judd B Kessler, and Abraham Othman. 2017. "Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation." *Operations Research,* 65(2): 314–336.

Budish, Eric, Peter Cramton, and John Shim. 2015. "The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response." *Quarterly Journal of Economics,* 130(4): 1547–1621.

Cespa, Giovanni. 2004. "A comparison of stock market mechanisms." *RAND Journal of Economics,* 803–823.

Chao, Yong, Chen Yao, and Mao Ye. 2017. "Discrete pricing and market fragmentation: A tale of two-sided markets." *American Economic Review,* 107(5): 196–99.

Chao, Yong, Chen Yao, and Mao Ye. 2019. "Why discrete price fragments U.S. stock exchanges and disperses their fee structures." *Review of Financial Studies,* 32(3): 1068–1101.

Chen, Daniel, and Darrell Duffie. 2021. "Market fragmentation." *American Economic Review,* 111(7): 2247–74.

Cramton, Peter. 2017. “Electricity Market Design.” *Oxford Review of Economic Policy*, 33(4): 589–612.

Daskalakis, Constantinos, Paul W Goldberg, and Christos H Papadimitriou. 2009. “The complexity of computing a Nash equilibrium.” *SIAM Journal on Computing*, 39(1): 195–259.

Demange, Gabrielle, David Gale, and Marilda Sotomayor. 1986. “Multi-item auctions.” *Journal of Political Economy*, 94(4): 863–872.

Duffie, Darrell, and Haoxiang Zhu. 2017. “Size discovery.” *Review of Financial Studies*, 30(4): 1095–1150.

Du, Songzi, and Haoxiang Zhu. 2017. “What is the optimal trading frequency in financial markets?” *The Review of Economic Studies*, 84(4): 1606–1651.

Friedman, Milton. 1960. “In defense of destabilizing speculation.” *Essays in economics and econometrics*, 133–141.

Gondzio, Jacek. 2012. “Interior point methods 25 years later.” *European Journal of Operational Research*, 218: 587–601.

Grossman, Sanford J, and Joseph E Stiglitz. 1980. “On the impossibility of informationally efficient markets.” *American Economic Review*, 70(3): 393–408.

Gul, Faruk, and Ennio Stacchetti. 1999. “Walrasian equilibrium with gross substitutes.” *Journal of Economic theory*, 87(1): 95–124.

Hatfield, John William, and Paul R Milgrom. 2005. “Matching with contracts.” *American Economic Review*, 95(4): 913–935.

Hatfield, John William, Scott Duke Kominers, Alexandru Nichifor, Michael Ostrovsky, and Alexander Westkamp. 2013. “Stability and competitive equilibrium in trading networks.” *Journal of Political Economy*, 121(5): 966–1005.

Hatfield, John William, Scott Duke Kominers, Alexandru Nichifor, Michael Ostrovsky, and Alexander Westkamp. 2019. “Full substitutability.” *Theoretical Economics*, 14(4): 1535–1590.

Kelso Jr, Alexander S, and Vincent P Crawford. 1982. “Job matching, coalition formation, and gross substitutes.” *Econometrica: Journal of the Econometric Society*, 1483–1504.

Klemperer, Paul. 2010. “The product-mix auction: A new auction design for differentiated goods.” *Journal of the European Economic Association*, 8(2-3): 526–536.

Klemperer, Paul D, and Margaret A Meyer. 1989. "Supply function equilibria in oligopoly under uncertainty." *Econometrica: Journal of the Econometric Society*, 1243–1277.

Kyle, Albert S. 1985. "Continuous auctions and insider trading." *Econometrica: Journal of the Econometric Society*, 1315–1335.

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

Kyle, Albert S, and Anna A Obizhaeva. 2016. "Market Microstructure Invariance: Empirical Hypotheses." *Econometrica*, 84(4): 1345–1404.

Kyle, Albert S, and Jeongmin Lee. 2017. "Toward a fully continuous exchange." *Oxford Review of Economic Policy*, 33(4): 650–675.

Kyle, Albert S, Anna A Obizhaeva, and Yajun Wang. 2018. "Smooth trading with overconfidence and market power." *Review of Economic Studies*, 85(1): 611–662.

Lahaie, Sebastien M., and David C. Parkes. 2004. "Applying Learning Algorithms to Preference Elicitation." *Proceedings of the 5th ACM Conference on Electronic Commerce*, 180–188.

Li, Sida, Xin Wang, and Mao Ye. 2021. "Who provides liquidity, and when?" *Journal of Financial Economics.*

Malamud, Semyon, and Marzena Rostek. 2017. "Decentralized exchange." *American Economic Review*, 107(11): 3320–62.

Mas-Colell, Andreu, Michael Dennis Whinston, and Jerry R Green. 1995. *Microeconomic Theory.* Vol. 1, Oxford University Press, New York.

McKenzie, Lionel W. 1959. "On the existence of general equilibrium for a competitive market." *Econometrica*, 54–71.

Mehrotra, S. 1992. "On the implementation of a primal–dual interior point method." *SIAM Journal on Optimization*, 2(4): 575–601.

Milgrom, Paul. 2000. "Putting auction theory to work: The simultaneous ascending auction." *Journal of political economy*, 108(2): 245–272.

Milgrom, Paul. 2009. "Assignment Messages and Exchanges." *American Economic Journal: Microeconomics*, 1(2): 95–113.

Nesterov, Yurii. 2004. *Introductory lectures on convex optimization: A basic course.* Kluwer Academic Publishers.

Nocedal, Jorge, and Stephen Wright. 2006. *Numerical optimization.* Springer Science & Business Media.

Ostrovsky, Michael. 2008. "Stability in supply chain networks." *American Economic Review*, 98(3): 897–923.

Parkes, David C., and Sven Seuken. 2018. *Economics and Computation.* Cambridge University Press.

Rostek, Marzena, and Ji Hee Yoon. 2020*a*. "Equilibrium theory of financial markets: Recent developments." *Journal of Economic Literature.*

Rostek, Marzena J, and Ji Hee Yoon. 2020*b*. "Exchange design and efficiency." Available at SSRN 3604976.

Rostek, Marzena J, and Ji Hee Yoon. 2020*c*. "Innovation in Decentralized Markets." Available at SSRN.

Sandholm, Tuomas, and Craig Boutilier. 2006. "Preference Elicitation in Combinatorial Auctions." In *Combinatorial Auctions.* , ed. Peter Cramton, Yoav Shoham and Richard Steinberg, Chapter 10. MIT Press.

Scarf, Herbert E, and Terje Hansen. 1973. *The computation of economic equilibria.* Yale University Press.

Tyc, Stephane. 2014. "A technological solution to best execution and excessive market complexity." *Quincy Data, LLC.*

Vandenberghe, L. 2010. "The CVXOPT linear and quadratic cone program solvers." UCLA. Available at `http://www.seas.ucla.edu/~vandenbe/publications/coneprog.pdf` Package documentation.

Vayanos, Dimitri. 1999. "Strategic trading and welfare in a dynamic market." *The Review of Economic Studies*, 66(2): 219–254.

Vohra, Rakesh V. 2011. *Mechanism Design: A Linear Programming Approach.* Cambridge University Press.

Wittwer, Milena. 2021. "Connecting disconnected financial markets?" *American Economic Journal: Microeconomics*, 13(1): 252–282.

Wunsch, Stephen. 1986. “Weekly Commentary.” The Financial Futures Department; Kidder, Peabody, & Co.

Yao, Chen, and Mao Ye. 2018. “Why trading speed matters: A tale of queue rationing under price controls.” *Review of Financial Studies,* 31(6): 2157–2183.

Zhang, Anthony Lee. 2020. “Competition and Manipulation in Derivative Contract Markets.” Available at SSRN 3413265.

Appendix

A Proofs

Proof of Theorem 3. It is clear from equation (11) that the order execution rate x_i and multipliers μ_i and λ_i may be chosen on an asset-by-asset basis, dramatically simplifying the problem, because the only connection between different orders operates through the price of the portfolio $p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi}$. In the primal problem, x_i is chosen to maximize utility. In the dual problem, the exchange chooses prices $\boldsymbol{\pi}$ and multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ to minimize the utility, taking into account that x_i will be chosen optimally. Differentiating the Lagrangian with respect to λ_i and μ_i, it is clear that if x_i were to be negative (because the portfolio price is so high that the buy order would want to become a sell order in the absence of an explicit constraints $0 \le x_i \le q_i$), then the exchange could reduce costs by increasing μ_i. Eventually, x_i is increased to the point where the constraint $x_i \ge 0$ is satisfied. Similarly, if the order would be overfilled due to low prices, the exchange would increase λ_i to lower costs, forcing the quantity x_i down to satisfy the constraint $x_i \le q_i$. Once both order size constraints are satisfied, it is not optimal for the exchange to adjust μ_i or λ_i further because doing so would increase the objective being minimized. When the constraints are strictly satisfied, $0 < x_i < q_i$, the exchange chooses $\mu_i = \lambda_i = 0$; it could reduce costs by choosing $\mu_i < 0$ or $\lambda_i < 0$ but this would violate nonnegativity constraints on the multipliers. Altogether, the optimal μ_i and λ_i chosen by the exchange satisfy the complementary slackness conditions $\mu_i x_i = 0$ and $\lambda_i(q_i - x_i) = 0$ and x_i stays within the required bounds $0 \le x_i \le q_i$. Thus, the order i contributes to the Lagrangian by choosing x_i to maximize $V_i(x_i) - x_i \mathbf{w}_i{}^\top \boldsymbol{\pi}$ subject to $0 \le x_i \le q_i$. This is exactly the value of order i's gains function $G_i(x_i^*)$ in equation (15), defined as the value $V_i(x_i)$ minus the cost $x_i p_i$ with $p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi}$. Thus, the Lagrangian is the sum of the gains functions $G_i(x_i^*)$, which is the desired result $G(\boldsymbol{\pi}) = \min_{\boldsymbol{\lambda},\boldsymbol{\mu}} \hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ subject to $\boldsymbol{\lambda} \ge \mathbf{0}$, $\boldsymbol{\mu} \ge \mathbf{0}$.

□

B Details Related to Optimization

This appendix provides a discussion of issues related to computation. Quadratic programming is well-studied. The results we rely on can be found in standard textbooks. This appendix describes standard optimization results in terms intended to make it easier for economists to understand the relationship between the economic problem and the optimization problem.

B.1 Efficiency of the Gradient Method

The gradient method has been studied extensively in the context of minimizing objective functions like our gains function. Textbook results about the gradient method show that if the objective function is well-enough behaved, it does allow market clearing prices (zero gradient) to be approximated arbitrarily closely in theory, but the number of iterations required is so large that the method is not fast enough for our purposes. Here we review these standard results.

The gradient method proceeds iteratively by adding to the most recent price an increment which is proportional to net excess demand. Describing a gradient method precisely requires specifying the proportionality constant by which the gradient is multiplied before adding it to the most recent price. In the field of optimization, there are many different approaches to determine an acceptable step size, including a constant small step size, a thorough line search, and adaptive step sizes which change with the number of iterations.

For our purposes, it is sufficient to note that a small enough step size $\epsilon > 0$ in the direction of the negative gradient, $\Delta\boldsymbol{\pi} = -\epsilon \cdot \nabla G(\boldsymbol{\pi})$ always generates an improvement if the gradient is nonzero and the objective function is continuously differentiable:

$$G(\boldsymbol{\pi} - \Delta\boldsymbol{\pi}) - G(\boldsymbol{\pi}) \approx -\epsilon \nabla G(\boldsymbol{\pi})^{\mathsf{T}} \nabla G(\boldsymbol{\pi}) < 0. \tag{46}$$

Even if the objective function is continuously differentiable, the gradient may change by a very large amount over a very small interval. This can make the maximum safe step size so small that the gradient method does not converge to an optimum.

To deal with these issues, optimization researchers makes the assumption that the first derivative of the objective function is not only continuously differentiable but also satisfies a Lipschitz condition $|\nabla G(\boldsymbol{\pi}+\Delta\boldsymbol{\pi}) - \nabla G(\boldsymbol{\pi})| < L|\Delta\boldsymbol{\pi}|$ for some Lipschitz constant L. Our gains function $G(\boldsymbol{\pi})$ is not only continuously differentiable but also satisfies a Lipschitz condition because the derivative is piecewise linear.

The Lipschitz condition bounds how fast the gradient can change over a small interval. This makes it possible to prove that a small step size exists which is large enough that convergence to an equilibrium is guaranteed. Guaranteed convergence is slow. Nesterov (2004, Corollary 2.1.2, p. 70) shows that the error can be reduced by a factor of two if the number of iterations is doubled.[20] This theoretical results implies that the gradient method can slow down as it approaches the equilibrium. In applications, the gradient method often does slow down in a manner consistent with this theory. Since we need accurate solutions for our problem, we need a stronger theorem to make the gradient method applicable.

[20] Nesterov (2004, Theorem 2.1.7, p. 61) shows there are badly behaved functions such that, for numbers of iterations less than about $N/2$, reducing the error by a factor of four requires doubling the number of iterations.

Intuitively, the gradient method should converge faster if the the objective function is "more convex." Convex optimization theory deals with this using the concept of strong convexity. The objective function is "strongly convex" if it remains convex if a positive definite quadratic form is subtracted from it.

While our objective function is not strongly convex, it can be made strongly convex by adding a positive definite quadratic form to it. In the context of our problem, adding a quadratic form corresponds to making the exchange a stabilizing market maker in the traded assets. If we impute to the exchange a quadratic gains function $\frac{1}{2}(\boldsymbol{\pi}-\boldsymbol{\pi}_0)^\top \boldsymbol{\epsilon}(\boldsymbol{\pi}-\boldsymbol{\pi}_0)$, the exchange's implied demand function is the derivative $\boldsymbol{\epsilon}(\boldsymbol{\pi}-\boldsymbol{\pi}_0)$, which is a linear function of prices. The exchange does not trade at all if $\boldsymbol{\pi}=\boldsymbol{\pi}_0$, buys assets or portfolios with cheaper prices, and sells portfolios with more expensive prices compared with $\boldsymbol{\pi}_0$. In practice, the exchange's no-trade price vector $\boldsymbol{\pi}_0$ might be the prices from the most recent auction, perhaps adjusted in a direction so that the exchange liquidates previously acquired inventories. The positive definite symmetric matrix $\boldsymbol{\epsilon}$ might be a diagonal matrix (with all diagonal elements strictly positive), implying that the exchange places stabilizing buy and sell orders at every price for every individual asset, providing different levels of liquidity to different assets.

It is another standard textbook result from optimization theory that, with strong convexity, the gradient method converges faster than when the derivative merely satisfies a Lipschitz condition. Nesterov (2004, Theorem 2.1.15, p. 70) shows that reducing the error by a factor of two requires no more than a constant number of iterations. The constant depends on the ratio of the Lipschitz constant to the smallest entry on a diagonal matrix Q. If the exchange's demand function provides very little liquidity to some asset, this constant number of iterations can theoretically be very large. Thus, while the gradient method—and therefore a variation on Walrasian tatonnement—may theoretically find market clearing prices for our problem, the algorithm may be too slow to be practical unless the exchange trades very actively as a market maker. If the exchange does trade actively enough as a market maker, convergence to an equilibrium can be very fast, but the exchange will dominate trading volume.

B.2 Interior point methods

Intuition suggests two strategies for speeding convergence.

First, if the objective function had a continuous second derivative (which ours does not have), Newton's method could be used. It is well-known that Newtons's method converges quadratically to an optimum if the starting point is close enough to the optimum, and the objective function has a continuous second derivative which does not change too fast. Quadratic convergence means that the number of digits of accuracy doubles with each iteration! Unfortunately, since the first derivative of our gains function $G(\boldsymbol{\pi})$ is piecewise linear, the second derivative (hessian), denoted $\nabla^2 G(\boldsymbol{\pi})$, is not always defined. When it is defined, the Newton step is proportional to the product of the inverse of the hessian with the negative gradient, $-\left(\nabla^2 G(\boldsymbol{\pi})\right)^{-1} \nabla G(\boldsymbol{\pi})$. One intuition for nevertheless making Newton's method work is to "smooth out" the derivative so that the hessian always exists.

Second, a Walrasian auctioneer, frustrated by attempts to find market-clearing prices based on local information about the derivative of quantities with respect to prices, might be tempted to look deep into the book of unexecuted orders to find pools of potential liquidity which do not show up in local gradient information. By using global information rather than local information only, more informative price changes may be proposed at each iteration, thus speeding up convergence.

Both of these intuitions are implicitly implemented by interior-point methods of optimization.

Interior point methods, developed in the 1980s and refined in the 1990s, are motivated by the idea of improving Newton's method so that it implicitly uses global information about the shape of the relevant functions. Newton's method converges locally particularly fast when the hessian itself satisfies a Lipschitz conditions. As (Nesterov, 2004, Section 4.1) points out, Newton's method—unlike the gradient method—is scale invariant in the sense that changing the units of measurement of prices (like changing some prices from U.S. dollars per share to dollars per 100 shares but not changing other price conventions) does not change the Newton iterations at all: When the Newton step is calculated, changes to the inverse hessian exactly cancel out changes in the gradient. The implications of the Lipschitz condition on the hessian do change when units are rescaled. To make a Lipschitz-like condition applicable in a scale invariant manner, interior point methods replace the Lipschitz condition on the hessian with the concept of "self-concordant functions." Given two vectors $\mathbf{x}$ and $\mathbf{u}$ and a function f, define the univariate function g by $g(t \,|\, \mathbf{x}, \mathbf{u}) := f(\mathbf{x} + t \cdot \mathbf{u})$. The function f is self-concordant if there exists a uniform bound L such that $|g'''(t)| < L|g''(t)|^{3/2}$ for all $\mathbf{x}$ and $\mathbf{y}$. This modified version of a Lipschitz condition on the hessian uses the norm of the hessian itself to scale units so that Newton's method works better globally.

Both of these intuitions are implicitly implemented by interior-point methods of optimization. Unlike Walrasian tatonnement, which solves the dual problem of finding optimal prices, the interior point method uses information from the primal problem to solve the KKT conditions.

Interior-point methods replace inequality constraints with penalty functions. In our problem, the standard log-barrier penalty function $-\bar{\nu}\log(\mathbf{x})$ (element-by-element) can approximate the inequality constraints $\mathbf{x} \geq \mathbf{0}$ for small $\bar{\nu} > 0$. Penalty functions keep all points $\mathbf{x}$ strictly in the interior of the inequality constraint set. This requires the starting point of the optimization algorithm to be a feasible interior point of the inequality constraint set (but not the equality constraint set). For our problem, the natural starting point is no-trade ($\mathbf{x} = \mathbf{0}$). This point is feasible but not an interior point because it lies on the boundary of the constraint (since it satisfies the inequality constraint $\mathbf{x} >= \mathbf{0}$ exactly).

To deal with this issue, we again allow the exchange to be a market maker, buying and selling small quantities with globally linear demands as before. This allows any $\mathbf{x}$ on the interior such that $\mathbf{0} < \mathbf{x} < \mathbf{q}$ to be feasible; the exchange can take into its inventory any uncleared quantities. The utility function corresponding to the exchange's gains function $\frac{1}{2}(\boldsymbol{\pi} - \boldsymbol{\pi}_0)^{\top} \boldsymbol{\epsilon} (\boldsymbol{\pi} - \boldsymbol{\pi}_0)$ is $\mathbf{y}^{\top} \boldsymbol{\pi}_0 - \frac{1}{2} \mathbf{y}^{\top} \boldsymbol{\epsilon}^{-1} \mathbf{y}$, where $\mathbf{y}$ is the vector of assets traded by the exchange. The primal objective function is changed by adding this utility to it.

The interior point method is applied to an optimization of the primal problem, not the gains function (dual problem). The new optimization problem differs from our original primal optimization problem in three ways: (1) It adds the exchanges utility function to the objective function; (2) it adds the exchanges quantities traded to the market-clearing condition, and (3) it replaces the inequality constraints with penalty functions.

The modified maximization problem is

$$\max_{\mathbf{x},\mathbf{y}} \left[\mathbf{x}^{\top} \mathbf{p}^{H} - \tfrac{1}{2} \mathbf{x}^{\top} \mathbf{D} \mathbf{x} + \mathbf{y}^{\top} \bar{\boldsymbol{\pi}} - \tfrac{1}{2} \mathbf{y}^{\top} \boldsymbol{\epsilon}^{-1} \mathbf{y} + \bar{v} \log(\mathbf{x})^{\top} \mathbf{1} + \bar{v} \log(\mathbf{q} - \mathbf{x})^{\top} \mathbf{1} \right] \qquad \text{subject to} \qquad \mathbf{W}\mathbf{x} + \mathbf{y} = \mathbf{0}. \tag{47}$$

If clearing prices are $\boldsymbol{\pi}^*$, the exchange acquires inventories $\boldsymbol{\epsilon}(\boldsymbol{\pi}_0 - \boldsymbol{\pi}^*)$. Note that the exchange's preferences are defined over assets while the customer orders define preferences over portfolios.[21]

Not many functions are self-concordant. Fortunately, both the quadratic objective and the log barrier function are self-concordant. This is all that we need for Newton's method works well on this problem for any $\bar{v}$. Interior point methods rapidly and accurately solve this problem by solving different version of the problem as $\bar{v} \to 0$ using Newton's method (Nesterov (2004, Chapter 3)).

Solving the problem using this approach is equivalent to solving the KKT conditions (27)–(30) with three changes: (1) Add the exchanges demand to the first-order condition. (2) Add the exchange's quantities traded to the market-clearing condition. (3) Replace the complementary slackness condition $\boldsymbol{\mu}^* \cdot \mathbf{x}^* = \mathbf{0}$ with $\boldsymbol{\mu}^* \cdot \mathbf{x}^* = \bar{v} \cdot \mathbf{1}$, and then let $\bar{v} \to 0$.

[21] Specifically, $\mathbf{y}$ is a vector of length N and $\boldsymbol{\epsilon}^{-1}$ is an $N \times N$ positive definite matrix while $\mathbf{x}$ is a vector of length I and $\mathbf{D}$ is an $I \times I$ positive definite diagonal matrix. It is straightforward to approximate the exchange's orders by treating them as single-asset portfolio orders which have very high $\mathbf{p}^H$ and very low $\mathbf{p}^L$. This would make the additional notation for the exchange's trading unnecessary. We keep it separate to make analysis of exchange trading more transparent.

Unlike the solution for our original problem, the solution for our revised problem is unique because the exchange always has partially executable orders in the market, and these orders define unique prices:

Theorem 8. *At the limit* $\bar{\nu} \to 0$*, the problem* (26) *has a unique solution for both optimal (market-clearing) quantities* $\mathbf{x}^*$*,* $\mathbf{y}^*$ *and for prices* $\boldsymbol{\pi}^*$*.*

Proof. The problem (26) can easily be converted into almost exactly the same form as the problem (10) when the exchange is treated as a customer, the exchange' orders for assets are modified to be orders for portfolios, and the vectors **x** and **y** are then stacked together. The only difference is that the lower limit price on the exchange's orders is close to minus-infinity and the exchange's execution rates are very large numbers (even though the actual execution rate is small because the orders are only barely partially executable relative to the lower limit price near infinity). Thus, Theorems 1 and 2 imply existence of unique market clearing quantities and possibly non-unique clearing prices. Since the exchange's orders are all partially executable and the matrix $\boldsymbol{\epsilon}^{-1}$ is positive definite, the unique quantities traded by the exchange can be uniquely inverted to obtain the prices at which the exchange trades the optimal prices. □

Trading by the exchange solves the tie-breaking problem. When there are no executable orders to make prices unique, the exchange can make price reasonable by submitting a demand schedule with willingness to trade at recent prices or other prices deemed reasonable.

B.3 Simulation Methodology

To examine numerical feasibility of flow trading, we simulate a book of orders for assets and portfolios, then use an optimization algorithm to calculated clearing prices. Our simulations implement a variation on the CVXOPT package in a Python environment. Both Python and CVXOPT are open-source. CVXOPT solves quadratic programming problems using cone methods, a variation of interior point methods. This method solves the KKT equations by replacing the complementary slackness equations (30) with the modified conditions

$$\boldsymbol{\lambda}^* \cdot (\mathbf{q} - \mathbf{x}^*) = \bar{\nu}, \qquad \boldsymbol{\mu}^* \cdot \mathbf{x}^* = \bar{\nu}, \qquad \bar{\nu} > 0, \qquad \bar{\nu} \to 0. \tag{48}$$

For a given $\bar{\nu} > 0$, the algorithm can easily enforce the interior-point intuition that all guesses for primal parameters **x** and dual quantities $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$ strictly satisfy the inequality constraints. The correct solution is obtained in the limit $\bar{\nu} \to 0$, which allows inequality constraints to be satisfied as equalities, consistent with the original problem.

In each iteration of the CVXOPT algorithm, the KKT equations are linearized and solved in a manner that pushes $\bar{\nu}$ towards zero. Since the KKT equations are essentially first-order conditions, solving the linearized system is a variation on Newton's method. At each iteration, a different linearized system of equations must be solved for updates to $\mathbf{x}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\lambda}$. This requires solving a different positive-definite linear system with $3I + N$ unknowns on each iteration. For typical problems, solving this system dominates computation time.

It may seem algorithmically wasteful to solve a different complex linear system for each iteration. Why not solve a system once and reuse it? The answer is that the values of the current multipliers contain information about which orders are "likely" to be executable. The algorithm uses the multipliers to attach an implicit weight to each order. The weight goes to zero when the algorithm anticipates the order will be either nonexecutable or fully executable and therefore will be irrelevant for price dynamics at the margin. The weight goes to one when the multipliers imply that the order is likely to be partially executable and there highly relevant to price dynamics at the margin. This is the algorithm's way of looking deep into the order book and using information about every order to guide price changes from one iteration to the next, unlike pure gradient and Newton methods which only use local information. As multipliers change every iteration, these weights also change every iteration. Therefore, solving the linear system with different weights each iteration is important if new relevant information about order executability is being implied by changes in multipliers.

The CVXOPT package gives the user a choice between using a default solver or defining a user-supplied custom solver. Our problem has two features which make a custom solver particularly desirable. First, the number of orders I is likely to be much larger than the number of assets N. Second, the matrix $\mathbf{D}$, which the algorithm assumes to be a positive-definite symmetric matrix, is in our case diagonal and therefore easily inverted. This allows us to use a custom solver which uses the explicit inverse of $\mathbf{D}$ to substitute out updates to the variables $\mathbf{x}$, $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$, reducing the number of variables in the linear system from $3I + N$ to only N. This dramatically speeds up the algorithm.

For example, our base-case simulation assumes $N = 500$ assets and $I = 30{,}000$ orders. Our user-supplied solver calculates a Cholesky decomposition of a 500×500 matrix, which is far faster than a prohibitively costly Cholesky decomposition of the $90{,}100 \times 90{,}100$ matrix using the default algorithm.

Issues of Algorithmic Implementation Modern packages for solving quadratic programming problems using the "cone-programming" version of interior point methods often try to solve as wide a variety of problems as possible. This requires verifying that all inequality constriants can be approximated with self-concordant barrier functions. This verification process can be a computationally time consuming part of the algorithm. It is not needed for our problem, which we know from the outset satisfies all needed self-concordance assumptions.

For our implementation, we got better performance by using the documentation of the CVXOPT algorithm by Vandenberghe (2010) to implement our own version of the quadratic programming algorithm in Python using the numpy and scipy packages for matrix and vector math rather than the actual CVXOPT package itself (which relies on different functions for matrix and vector math). Our implementation gave almost identical results but was faster and more configurable. Our implementation is much simpler than the CVXOPT package itself because it is specialized to our specific quadratic program, with $\mathbf{D}$ a diagonal matrix and simple Euclidean cone constraints of the form $\mathbf{0} \leq \mathbf{x} \leq \mathbf{q}$ known ex ante to be approximated with lob-barrier functions satisfying self-concordance assumptions.

Intuition suggests using the solution from the previous second's auction to "warm start" a search for the solution for the next second's auction. One disadvantage of interior point methods in general and the CVXOPT algorithm in particular is that the parameter $\bar{v}$ must also be reset when the optimization is restarted. This affects the appropriate starting values for multipliers and mak difficult to warm-start the algorithm.

An important issue concerning algorithmic efficiency concerns whether various calculations can be parallelized. Most of the computations are matrix-matrix or matrix-vector products in which the matrix is sparse. Sparsity arises because the portfolios being traded are typically single assets or pairs trades, which can obviously be described in a sparse manner. While the sparse matrix multiplication can theoretically be parallelized across orders, our Python environment may not have exploited this possibility efficiently.[22]

Developing efficient parallel algorithms is an active area of computation research and may lead to future improvements. Therefore, in a future production environment, we expect that increases in computation time related to increases in the number of orders will not be a big computation bottleneck because more cores can be used to perform computations which aggregate information across orders.

The algorithm also requires many Cholesky decompositions (or other matrix decompositions), which are not easily parallelized. Increasing computation time related to increasing the number of assets may prove to be more challenging bottleneck because adding more computing resources may not speed up the $O(N^3)$ operations required for a Cholesky decomposition as easily.

[22] We used the sparse_dot_mkl package for sparse matrix operations. This package calls parallelized functions from the Intel Math Kernel Library (MKL) but did not offer much improvement in execution speed on multiple cores, for reasons we do not yet fully understand.

B.4 Simulation Details

Our simulated order book tries to make the optimization problem difficult for the same three reasons that the gradient method is impractical.

1. Scaling: The size of the market varies cross-sectionally across assets and portfolios. Some assets have a large number of orders, with the average order being large in dollar size. Other markets have a small number of orders of small size.

2. Portfolio orders: The simulated order book orders for assets, orders from equally-weighted and value-weighted market portfolios, orders for equally-weighted and value-weighted industry portfolios, and pairs-trade orders to swap one asset, market portfolio, or industry portfolio for another. These portfolio orders make the hessian for the gains function poorly conditioned. In particular, the differences between equally-weighted and value-weighted portfolio orders introduces a potential difficulty for an optimizer.

3. Changing hessian: To make the hessian change dramatically as prices change, the average difference $\mathbf{p}^H - \mathbf{p}^L$ can be made small. When $\mathbf{p}^H - \mathbf{p}^L$ is large, market depth does not change much as prices changes, the problem looks linear to an optimizer, so clearing prices are likely to be easy for the optimizer to find. When $\mathbf{p}^H - \mathbf{p}^L$ is small, the range over which an order is partially executable small, and the hessian changes greatly when prices change a small amount. In the limit $\mathbf{p}^H - \mathbf{p}^L \to 0$, the hessian itself goes to zero for a randomly guessed price vector. Gradient methods and Newton methods, which rely only on local derivative information, cannot make good guesses about search direction because they have no information about prices where liquidity is located in the limit order book.

Since interior point methods are designed to deal with all of these problems, we expect the optimizer to be able to calculate clearing prices when liquidity varies greatly across assets, there are many equally- and value-weighted orders for market and industry portfolios, and orders have small ranges over which they are partially executable.

The problem difficulty can also be varied by varying the intensity of exchange trading. The problem is easier when the exchange provides a deeper market for all assets. In our simultations, we try to keep exchange trading as small as possible, consistent with numerical stability of the algorithm.

The order book is constructed using parameters listed in Table 2, which provides a name for the parameter, a base-case assumption for the parameter value, and a description of the parameter. The number of assets is 500. There are 10,000 orders for individual assets, 10,000 orders for various indexes, and 10,000 orders which randomly buy an index or asset and sell an equal expected dollar value of another asset or index asset. In addition to value-weighted- and equally-weighted market indexes, there are also 5 indexes for dollar volume quantiles and 10 indexes for industries. Assets are assigned to volume and industry indexes by sorting the assets by expected

Table 2: Parameters for simulating an order book. The value of each parameter is the base-case assumption.

Variable	Value	Description
N	500	number of assets
MA	10,000	number of orders for individual assets
MX	10,000	number of orders for indexes
$M2$	10,000	number of pairs-trade orders to swap assets or indexes
$num\_quantile\_indexes$	5	number industry quantiles
$num\_industry\_indexes$	10	number of industries
ew_index_share	0.05	dollar volume share of equally-weighted index orders
$vw\_quantile\_index\_share$	0.08	dollar volume share of value-weighted index orders
$ew\_quantile\_index\_share$	0.02	dollar volume share of equally-weighted index orders
$ew\_industry\_index\_share$	0.08	dollar volume share of equally-weighted index orders
$ew\_industry\_index\_share$	0.02	dollar volume share of equally-weighted index orders
$index\_price$	\$100.00	price of one share of any index
$std\_num\_orders\_asset$	1.7	log-standard deviation of dollar order size for given asset
$mean\_asset\_price$	\$100.00	mean expected price of each asset
std_asset_price	0.00	log-standard deviation of asset price
$invariance\_exponent$	1/3	invariance exponent
$invariance\_c$	\$1.00	invariance constant
$invariance\_m$	0.60	invariance moment ratio
std_order_size	1.5	log-standard deviation of dollar order size for given asset
std_limit_price	0.10	log standard deviation of order's limit price midpoint
$fraction\_buy\_orders\_asset$	0.50	fraction of asset orders which buy an asset
$avg\_ph\_minus\_pl\_bp$	1.00	$2*(p_i^H - p_i^L)/(p_i^H + p_i^L)*10^{-4}$, $p_i^H - p_i^L$ in basis points
$std\_ph\_minus\_pl$	2.50	log standard deviation of $p_i^H - p_i^L$
$frac\_exchange\_liquidity$	10^{-8}	larger value implies exchange provides more liquidity

trading volume, then assigning the assets to value quantiles by their rank in the sort and assigning assets to "industry" quantiles according to their rank modulo the number of industry indexes. The dollar-volume shares of the equally-weighted index orders, volume-weighted quantile index orders, equally-weighted quantile index orders, volume-weighted industry-index orders, and equally-weighted industry-index orders are 0.05, 0.08, 0.02, 0.08, and 0.02, respectively; the remaining 0.75 share of index volume is volume-weighted index orders.

Positive, continuous random variables are assumed to have a log-normal distribution. All index prices and asset prices are normalized to have an expected price of \$100.00 and log-variance of zero. The matrices and vectors defining the problem are scaled in the algorithm so that each iteration is unaffected by changes in expected stock prices (which corresponds to stock splits). The expected price is defined as the mean of the order midpoint $(p_i^H + p_i^L)/2$. There is no concept of fundamental value; prices depend only on orders.

We use the market microstructure invariance assumptions of Kyle and Obizhaeva (2016) to convert assumptions about the dollar volume all orders if fully executable into assumptions about the number and dollar size of orders. For any asset or index, the dollar volume of orders is divided into a number of orders that grows with the two-thirds power of dollar volume, which implies that the size of each order grows with the one-third power of dollar volume. We use Kyle–Obizhaeva calibration of a dollar invariant constant and moment ratio to calibrate the expected size of orders. This calibration does not mean much in the context of a one-second auction; it does not affect results about computation time either. For a given asset or index, the dollar size of orders has a log-standard deviation of 1.5, consistent with empirical results of Kyle–Obizhaeva.

The cross-sectional distribution of expected number of orders across assets is assumed to have a log-standard deviation of 1.7. This large standard deviation creates a few assets with many orders and many illiquid assets, some of which may have no orders at all. Thus, a small amount of exchange trading is helpful in pinning down prices for assets which might otherwise only be traded as components of indexes.

We assume that the log-standard-deviation of the order midpoint is 10 percent. Half the orders are expected to be buys and half are expected to be sells.

The average difference between p_i^H and p_i^L is assumed to be 1.00 basis point. For a typical order to trade an asset of portfolio with a price of \$100.00, this assumption implies that we expect $p_i^H = \$100.005$ and $p_i^L = \$99.995$. This calibration is designed to stress the algorithm, not to capture realistic expectations about market operation. We expect optimal trading strategies to make the difference between p_i^H and p_i^H much larger in order to limit price impact and profit from other traders' price impact.

Finally, there is a difficult-to-interpret parameter, which defines how aggressively the exchange trades each asset. A larger value of this parameter results in the exchange market-making orders capturing a larger fraction of volume. The baseline value of 10^{-8} leads to little trading by the exchange.

Altogether, these assumptions attempt to create a difficult optimization problem by having huge variation in liquidity across assets, including assets with no orders; poorly conditioned matrices resulting from index orders, with even worse conditioning due to mixing equally- and value-weighted indexes; equally-weighted- and value-weighted industry indexes which create poor conditioning along different dimensions from volume-based portfolios; small differences between p_i^H and p_i^L; and very little exchange trading to stabilize the problem.

An important algorithmic implementation detail concerns the manner in which portfolios are aggregated. If there are thousands of orders to trade a particular portfolio, such as the value-weighted index, instead of multiplying by the same portfolio weights thousands of times, the algorithm first adds up the portfolio demands for the particular portfolio across orders, then multiplies by the portfolio weights only one time. Our algorithm does this by expressing the matrix of portfolio weights **W** as the product of a sparse matrix defining a “master list” of allowed portfolio weights and a matrix of order information which identifies one or two portfolios in the master list, depending on whether the order is a pairs trade or not. These computations involve very sparse matrices because both the master list specifications and the order information for individual assets and pairs trades are sparse. Sequencing the sparse matrix multiplications efficiently improves performance dramatically because multiplying portfolio weights by orders’ execution rates is already a computational bottleneck.