Flow Trading*

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

April 22, 2022

Abstract

We propose a new market design for trading financial assets. The design combines three elements: (1) orders are downward-sloping linear demand curves with quantities expressed as flows; (2) markets clear in discrete time using uniform-price batch auctions; (3) traders may submit orders for portfolios of assets, expressed as arbitrary linear combinations with positive and negative weights. 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 arbitrary portfolios. Clearing prices and quantities are shown to exist, with the latter unique, despite the wide variety of preferences that can be expressed via portfolio orders. Calculating prices and quantities is shown to be computationally feasible. Microfoundations for portfolio orders are provided. The proposal has six advantages over the current market design. Flow trading (1) eliminates sniping and the speed race, (2) avoids the complexities and inefficiencies caused by tick-size constraints, (3) reduces the cost and complexity of trading large quantities over time, (4) reduces the cost and complexity of trading portfolios, (5) reduces the cost and complexity of providing liquidity in correlated assets, and (6) improves fairness and transparency of optimal execution.

*We thank Elizabeth Baldwin, Paul Klemperer, Paul Milgrom, David Parkes, and Marzena Rostek for helpful conversations. Disclosures: Budish is an advisor to a project pursuing frequent batch auctions for decentralized finance. Cramton consults on market design and is an academic advisor to Carta on the design of a private equity exchange. 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 under the European Union's research and innovation program (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 status-quo design Current exchanges for trading equities and many other financial assets, such as futures, options, and treasury bonds, 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 or better," which has one maximum quantity and one limit price. The orders are processed continuously, one at a time in order of arrival, with incoming "executable" orders matched in whole or in part with "non-executable" orders resting in the limit order book. Orders are for single securities rather than for portfolios of securities. Displayed bids and offers respect a minimum "tick size," which is typically $0.01 per share for U.S. stocks, and a minimum "lot size," which has historically been 100 shares for most U.S. stocks. Traded quantities must also respect a minimum tick size and minimum lot size.

This market design is the natural electronic version of the limit order books used in the era of human trading—tracing not only to the era of specialists and trading pits but also back to trading under the buttonwood tree. A human can run a limit order market with pen-and-paper or simple electronic recordkeeping if the orders arrive slowly enough, and computers can run limit order markets at modern speeds and order volumes.[1]

However, there are multiple ways that this market design creates unnecessary costs, complexity, and perceptions of unfairness for investors and other financial market participants, especially given the market design possibilities afforded by modern computers.

First, since any order resting on the limit order book is subject to immediate execution against the next incoming order, any time there is new public information that affects an asset's market price, resting limit orders risk being "picked off" or "sniped" by high-frequency traders acting on this information. Such orders trade at a stale price. This sniping raises the cost of providing liquidity using limit orders and is perceived by many market participants to be unfair. Recent evidence suggests that such sniping races constitute over 20% of all trading volume and constitute from 17% to 33% of the market's cost of liquidity depending on the measure used (Aquilina, Budish, and O'Neill (2022)).

Second, the discrete minimum tick size, which is necessary to prevent an explosion

[1] See MacKenzie (2021) for a history of this evolution from human-based trading to computer-based trading. See Aquilina, Budish, and O'Neill (2022, Section 2) for a detailed overview of the market design and associated computer systems architecture for handling modern levels of speed and order activity.

of message traffic under the current market design, artificially constrains the market's cost of liquidity. This constraint has been shown to lead to (i) high-frequency trading races for queue position, (ii) a proliferation of complex order types to navigate this race for queue position, and (iii) the proliferation of exchanges with creative fee schedules designed to circumvent this constraint. Both sniping and tick-size constraints also likely play a role in the proliferation of off-exchange trading, now nearly 50% of equity volume in the United States.[2]

Third, institutional investors trading large quantities of stock typically now do so by placing and canceling thousands of small orders spread out over time to reduce price impact and disguise trading intentions. This was anticipated by Black (1971). If an institutional investor's trading leaves a detectable statistical trace, algorithmic trading firms who detect the trading demand can profitably trade in front of the institutional investor.[3] Institutional investors therefore must have access to complex, expensive trading platforms to manage their orders, or risk being algorithmically front run. Such trading tools are unavailable to many smaller investors.

Fourth, these costs and complexities of optimal trading are even more severe for investors trading portfolios or engaging in long-short arbitrage strategies. Not only must investors manage price impact and smoothing their trading over time for each individual security in the trading strategy, but they must also handle the additional complexity that comes from managing the relative rates of trade across the different securities in the trading strategy. Some indirect evidence on the value of efficiently trading portfolios comes from the rise of exchange traded funds (ETFs). ETFs are redundant assets that enable investors to trade portfolios efficiently, in exchange for a management fee on holdings that averages about 20 basis points. ETFs now constitute a remarkable 40% of all U.S. stock market volume.[4]

[2]See the series of papers Chao, Yao, and Ye (2017, 2019), Yao and Ye (2018), and Li, Wang, and Ye (2021) on the various complexities created by tick-size constraints in U.S. equity markets, with additional references contained therein. Data on the share of off-exchange trading is available from SIFMA and was cited in Sept 2021 Senate testimony by SEC Chair Gary Gensler.

[3]As one simple example, Hasbrouck and Saar (2013) point out that execution algorithms that trade every second, on the second, leave an obvious statistical trace in a continuous-time market. If trading can take place at any nanosecond, it would be an astonishing coincidence for a sequence of trades to occur at exactly 1.000000000, 2.000000000, 3.000000000, etc. Note that the same trading would not leave as obvious a statistical trace in a discrete-time batch process market, in which all trade occurs at exactly 1, 2, 3, etc.

[4]ETF volume is computed from CRSP (ETFs are share code 73). The 40% figure is ETFs' proportion of on-exchange trading volume in dollars. Vanguard reports that the asset-weighted average ETF expense ratio for non-Vanguard ETFs is 0.24% in 2020.

Flow trading This paper proposes a new market design for financial exchanges, "flow trading." The design is motivated by the costs and complexities described above and the design possibilities enabled by modern computational power.

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 the Status Quo Design

Flow trading is a combination of three key elements (Table 1). First, instead of limit orders that define demands as step functions of price, traders place flow orders that specify demands as piecewise-linear downward-sloping functions of price, with quantities expressed as flows rather than as discrete quantity changes (Kyle and Lee (2017)). For example, "Buy a maximum of one share per second at $150.00 or better, declining to zero shares per second at $150.10 or worse, until 1000 shares are bought."

Second, instead of the market processing orders one at a time in sequence, orders are processed in discrete time using uniform-price batch auctions ("frequent batch auctions," Budish, Cramton, and Shim (2015)). Suppose the discrete-time interval is one second. A flow order to buy at a maximum rate of one share per second will buy one share per batch if fully executable at the clearing price, a fraction of a share per batch if partially executable (i.e., the clearing price is in the range where the order's demand is strictly downward sloping), or no shares per batch if non-executable. Orders persist over many auctions. An order remains outstanding until the trader cancels it or a user-defined termination criterion is met, such as the cumulative purchase of 1000 shares.

The combination of flow orders and batch auctions allows prices and quantities to be approximately continuous—tiny fractions of shares can trade each second within a nearly continuous price grid. For example, quantities could be expressed in nano-shares (billionths of shares) and prices in micro-dollars (millionths of dollars). 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. In our proposed design, prices and quantities can be approximately continuous without issue.

Third, instead of restricting to orders for single assets, our design allows traders to directly trade arbitrary portfolios of assets. In our design, a "portfolio" is any user-defined linear combination of assets, in which asset weights can be arbitrary positive or negative real numbers. Market clearing prices balance the demand and supply for each asset on an asset-by-asset basis, not on a portfolio-by-portfolio basis. The number of market clearing constraints is equal to the number of underlying assets, not the number of different portfolios—composed from these assets—for which orders are placed. Thus, a specific portfolio order may be matched against multiple orders for individual assets and other, different portfolios. Portfolio orders allow assets to be either complements or substitutes. If two assets 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 weights with opposite signs, 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 buy the S&P 500 has positive weights on each of the 500 stocks in the index. An order to sell a portfolio of assets, which represents an upward-sloping supply curve for the portfolio, is implemented as an equivalent downward sloping demand curve for a portfolio with negative weights on the assets being sold and zero weight on all other assets.

If the underlying space of assets is redefined by rotating assets with a nonsingular linear transformation, the set of portfolios a trader can trade is unaffected by such a change in basis assets.

In sum, relative to the status quo market design, the proposed market design makes time discrete instead of continuous, prices and quantities continuous instead of discrete, and allows participants to directly trade arbitrary user-defined portfolios.

Benefits Flow trading directly addresses the four concerns raised above about the status quo market design. First, sniping is directly addressed by discrete-time batch processing (Budish, Cramton, and Shim (2015)). Moreover, flow trading makes the executed quantity proportional to the length of time, which means that even if new public information arrives just before the next batch auction, so that regular traders are vulnerable to sniping, the actual quantity executed at unfavorable prices will remain small. Sec-

ond, the complexities and inefficiencies caused by tick-size and lot-size constraints are directly addressed by making prices and quantities approximately continuous. There no longer would be a reason to use non-standard exchange fee schedules or off-exchange trading venues to "hack the penny." Additionally, there no longer would be an incentive to race for advantageous queue position, further reducing the rents from speed. Third, investors who wish to trade large quantities over some time can do so directly, with a single order. They can easily tune the urgency of trade by choosing the maximum flow rate— trading more slowly if their information is not time-sensitive and vice versa. In effect, the ability to trade at the time-weighted average price (TWAP) is built directly into the market design. Moreover, since trading is batched, it is easier for a large trader to blend in with other traders (as in models such as Kyle (1989), Vayanos (1999), Kyle, Obizhaeva, and Wang (2018), Du and Zhu (2017)) without complex infrastructure. Fourth, investors who wish to trade portfolios can do so directly. Investors can define and directly trade their own custom ETFs, or long-short arbitrage portfolios, etc. Again, this reduces the need for costly trading infrastructure—expensive for large investors and unavailable to many smaller investors.

Another benefit of flow trading, related both to this last point about trading portfolios and to arguments by Budish, Cramton, and Shim (2015), is that market participants can more easily provide liquidity across correlated assets and 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. With 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). Even if an investor arrives wanting to buy just A, the order is automatically incorporated into A and B clearing prices. 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.

Last, the new market design significantly 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 feature allows every trader, whether trading 100 shares or 100,000, to infer exactly the execution

rate on their order from publicly announced clearing prices. An institutional investor trading a sophisticated portfolio can confirm they received the correct execution. This ability perhaps does not sound radical, but it is a significant transparency improvement over the current market design, where checking whether one's order received appropriate execution is 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 (see Rostek and Yoon (2020) for a recent survey of these issues). 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. Flow trading removes some of the unnecessary technological costs and complexities surrounding this game, but the fact remains that large trades will move prices.

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 microfoundations 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 impute a quasi-linear quadratic utility function to 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 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 since it satisfies both sets of constraints. This setup allows us to use known results from convex optimization to prove existence of unique equilibrium quantities.

Equilibrium prices are 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 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.

To show computational feasibility of the market design, we start by showing our problem has a structure such that the gradient method (equivalent to Walrasian 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 may 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 the clearing prices. This allows us to use interior point methods, which are 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 clearing prices for a simulated order book using our own implementation of a public-domain interior-point method on an ordinary workstation. In a market with 500 assets and 100,000 orders, our algorithm calculates prices in about 0.15 seconds in the baseline scenario (with the computation time ranging from 0.12 to 0.27 seconds when we consider a wide range of parameter values). With 500 assets and 1,000,000 orders, computation time is about 0.56 seconds. With 2000 assets and 100,000 orders, the computation time is about 1.1 seconds. 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 the solution to the problem is in P and not NP.

We provide a stylized microfoundation 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 de-

mands can be implemented with downward-sloping portfolio orders consistent with our proposal. This asset-rotation approach works because the variance penalty in a CARA-normal setting generates a positive-semidefinite covariance matrix, which makes portfolio demands downward-sloping. The approach generalizes to taking account of linear price impact under the mild assumption that trading any portfolio has a positive (quadratic) price impact cost.[5]

Structure of the paper The rest of the paper is structured as follows. Section 2 discusses related literature. Section 3 describes flow orders. Section 4 discusses the existence and uniqueness of market clearing prices and quantities. Section 5 shows computational feasibility of our proposal. Section 6 provides a microfoundation for portfolio orders. Section 7 discusses implementation and policy issues. Section 8 concludes.

2 Related Literature

We divide our discussion of related literature into two parts. Section 2.1 discusses the prior work related to the flow trading market design. Section 2.2 discusses prior work that is related to the results that equilibrium prices and quantities exist.

2.1 Literature Related to the Proposed Market Design

The 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. 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.

The two prior works most closely related to our paper are 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

[5] 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. Moreover, we then find that a trader who wishes to use $K' < K$ orders optimally does so by sorting on the portfolio Sharpe ratios, which may be practically appealing as well.

for financial exchanges. Combining these two market design ideas yields a market design for financial assets in which time is discrete instead of continuous, and prices and quantities are continuous instead of discrete. This is appealing for many reasons described above. Put another way, the present paper shows that these two prior market design ideas are complements, not substitutes.

The third ingredient of the market design proposal, portfolio orders, is novel to this paper. To be more precise, 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. We suggest via a long list of example use cases that our proposed language is practically useful for real-world financial markets. Different ways of representing preferences for portfolios also might not yield the existence and computability results we obtain here.

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 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.

Another closely-related body of work by Li, Wang, and Ye (2021), Chao, Yao, and Ye (2019), Chao, Yao, and Ye (2017) and Yao and Ye (2018) 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, our market design makes time discrete and prices continuous, thus eliminating the inefficiencies caused by tick-size constraints.

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 im-

pact 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 distributions 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 allowing orders to trade one asset to be contingent not only on the asset's price but on the prices of other assets (Cespa (2004), Rostek and Yoon (2021), Wittwer (2021)). For example, Rostek and Yoon (2021) study strategic behavior in a market with multiple assets and imperfectly competitive traders, under market designs with both contingent and non-contingent orders. In their framework, contingent orders allow a trader's demand for one asset to be a function of the price of all other assets, whereas non-contingent orders require that a trader's demand for each asset is a function only of the price of that asset. Portfolio orders, as defined here, are an intermediate case between contingent and non-contingent orders. A trader's demand for one asset can depend contingently on the prices of other assets, but only through these other asset prices' effects on the price of the trader-defined portfolio the assets belong to. In Section 6 we show that traders can use a collection of portfolio orders to implement their optimal fully-contingent demand.

Surprisingly, in these models, which study a one-shot Walrasian auctioneer framework, the efficiency and welfare consequences of allowing for contingent demands are ambiguous. Each trader's individually optimal demand is indeed contingent on all asset prices, but allowing for contingent orders affects incentives to strategically shade demand and supply, and the net effects of this incremental strategic behavior can affect efficiency or welfare in either direction. Chen and Duffie (2021) provide a related insight by studying fragmentation of trade of the same asset across multiple trading venues.[6] We note that these analyses do not study the efficiency issues which motivate this paper's market design proposal, such as reducing the technology and intermediation costs of trading portfolios (in these models, trading, including complex trading, is free) or re-

[6] On the other hand, Antill and Duffie (2020) find that fragmentation of trade of the same asset across multiple trading venues is unambiguously *negative* for efficiency in the case where one trading venue is the center of price discovery and other trading venues engage in size discovery (i.e., trade at the price discovered by the price-discovery venue).

ducing latency arbitrage opportunities across venues (in these models there is mostly just a single trading period). We also emphasize in the conclusion that extending this style of analysis to the case of portfolio orders as introduced here is an interesting open question for research.

2.2 Literature Related to Existence Results

We obtain existence of market-clearing prices and quantities despite the wide range of preferences, including both substitutes and complements, that can be expressed using portfolio orders. In this subsection we describe the relationship of our existence results to the textbook general equilibrium theory approach and to the literature on indivisible goods.

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. 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. 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, making the second derivative of the planner's objective function negative (semi) definite. 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.

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.

Kelso Jr. and Crawford (1982) 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, Kominers, and Westkamp (2021) discuss the relationship among many of these criteria and provide 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 Baldwin and Klemperer (2019), or 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 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 deserve special mention are Klemperer (2010), which proposes the product-mix auction, and Milgrom (2009), which proposes the assignment auction. (These papers in turn descend from Shapley and Shubik (1971) and 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 versus our proposal is the preference language. In our design, participants have piecewise-linear demands for portfolios of assets, which can have arbitrary user-defined positive and negative asset weights. In Klemperer's and Milgrom's designs participants have piecewise-constant demands, expressing preferences over mutually-exclusive substitutable assets, including Shapley–Shubik unit-demand for substitutes preferences as a special case. For example, our design would allow a user to buy a portfolio consisting of assets A and B in some user-specified ratio, with downward sloping demand for the portfolio, whereas Klemperer's and Milgrom's auction designs would allow the user to buy a specified quantity of whichever of A or B gives them more surplus at realized prices. This difference in language then drives differences in the statements and methods of proof for existence and uniqueness results. Practically, the papers 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 one-shot, high-value allocation problems—e.g., a high-value auction for toxic assets during the financial crisis, or a spectrum auction.

3 Flow Orders

3.1 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 step-wise demand curve, with full demand (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, $\mathbf{w}_i \coloneqq (w_{i1}, \ldots, w_{iN})^\top$, 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 the asset is not a part of that portfolio.

2. *Flow orders specify piecewise-linear downward-sloping demands.* Each order specifies two prices: a lower limit p_i^L and an upper limit p_i^H, with $p_i^L < p_i^H$. 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. It 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 interpolates the quantity demanded from full quantity at p_i^L to zero quantity at p_i^H.[7] 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 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 now), users can tune their urgency to trade.

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

[7] 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 the lower limit p_i^L, and demand zero at prices weakly worse than the upper limit p_i^H.

Next, we define a flow order's demand within a batch auction. Assume 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 $Q_i^{\max}$.[8] 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 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 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, 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 equa-

[8] 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 $Q_i^{\max} - Q_i^t$, and increase p_i^L so that the slope of the demand curve is the same as it was originally.

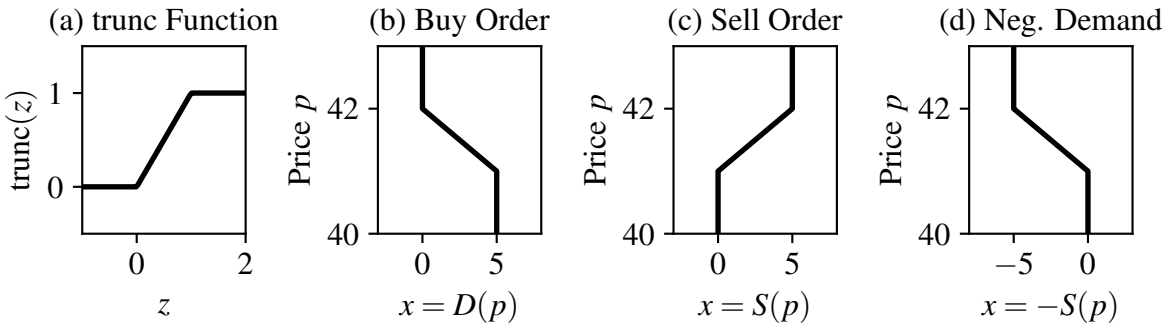

Figure 1: Plots of (a) the function trunc(z); (b) a single buy order, with pricing parameters p_i^L = \$41.00 and p_i^H = \$42.00, and maximum flow demand of q_i = 5.00 portfolio units per batch auction; (c) a single sell order, initially plotted as an upward-sloping supply curve with one upward-sloping 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.

tion (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 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.

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

First, observe that while the above demand function in equation (2) has a single

downward-sloping segment, the user can define an arbitrary piecewise-linear downward-sloping demand function for a given portfolio with multiple flow orders.

Second, order specification using the tuple of parameters $(\mathbf{w}_i, p_i^L, p_i^H, q_i, Q_i^{\max})$ contains an intentional redundancy of notation. Buying a portfolio containing one share each of two stocks at a rate of ten portfolio units per batch auction is equivalent to buying a portfolio containing half a share of each stock at a rate of twenty portfolio units 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. We do this because in some circumstances it will be natural to normalize some stocks' individual weights to one or minus one, while in others it may be more natural to normalize the sum of weights.

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 an example of such a proxy instruction: cancel the order from the market once the cumulative total quantity $Q_i^{\max}$ has been reached. Another example is time-in-force instructions, such as "good for day" or good for some other specified period. 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.

3.2 Examples

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

1. Standard limit order.

   A standard limit order expresses preferences to buy or sell a specified quantity of one asset at one limit price. A flow order can be specified to approximate a limit order. First, when only one weight w_n is nonzero, the order is an 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, $Q_i^{\max}$. 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 possible to p_i^L. Theoretically, we obtain a 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 essentially guaranteed to execute.[9] 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 period).

3. Pairs trades.

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

4. Portfolio trades.

   A portfolio trade is 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.

   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 index portfolios by adjusting portfolio weights—e.g., adjusting weights based on valuation models or setting to zero weights for assets that fail a screening criterion such as environmental, social, and governance criteria.

[9] 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.

5. General long-short strategies.

   A general long-short strategy combines the previous two cases: multiple positive and 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, 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 $p_i^L = -\$42.00$, $p_i^H = -\$41.75$

3.3 Limitations of the Language

There are 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 4.2).

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 take advantage of this linearity. We view the linearity restriction as a less important limitation 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 approximate such preferences with marketable orders if prices are sufficiently continuous.

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 approximate such preferences with marketable orders if prices are continuous enough.

4 Market Clearing Prices and Quantities

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

4.1 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 $p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi}$. Then, convert the demand for portfolio units $D_i(\mathbf{w}_i{}^\top \boldsymbol{\pi})$ into the demand for individual assets by multiplying by the portfolio weights $\mathbf{w}_i$. Last, 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:

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

The function $D(\cdot)$ maps asset price vectors $\boldsymbol{\pi} \in \mathbb{R}^N$ to net asset quantity vectors $\mathbf{q} \in \mathbb{R}^N$. A price vector is market clearing if each asset's net excess demand is zero:

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

This market clearing condition defines N equations in N unknowns. At clearing prices $\boldsymbol{\pi}$, order i's trading rate for the individual assets is given by $D_i(\mathbf{w}_i{}^\top \boldsymbol{\pi})\mathbf{w}_i$ (i.e., by its demand for portfolio units at the clearing prices times the portfolio weights).

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

4.2 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.

An order's demand is a linear function of prices in the range of prices where the order is partially executable. Therefore, a quadratic quasilinear utility function defines preferences. The constraints preventing overfilling or underfilling the order are linear inequality constraints. Market clearing consists of linear equality constraints. Putting this together results in a quadratic program—maximizing a quadratic objective function subject to linear constraints.

Quadratic programs have been thoroughly studied and are well-understood. Given the structure of our problem, we can use well-known results to show 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.

Imputing utility functions to orders is a convenient mathematical modeling device. We proceed as though orders directly represent traders' 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. Rather, the method provides a practical approach to finding clearing prices and quantities consistent with bids.

Pseudo-Utility Let $V_i(x_i)$ denote the dollar utility of order i from a trade rate of x_i in portfolio units per second. To find $V_i(x_i)$, we first define the marginal utility function $M_i(x_i)$ 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 $x_i \in [0, q_i]$ into prices $p \in [p_i^L, p_i^H]$.[10] Rearranging equation (2) we have:

[10] 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.

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

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

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

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

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

We will think of $V_i(x_i)$ as defined for all $x_i \in \mathbb{R}$, with order specifications imposing the constraint $x_i \in [0, q_i]$.[11]

Value Maximization Our problem of finding clearing prices is formulated as two optimization problems, a primal problem of finding quantities that maximize as-bid dollar value and a dual problem of finding prices that minimize the cost of non-clearing prices. The first-order conditions for optimality 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 value, defined as the sum of pseudo-utility functions across orders,

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

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.

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

Indeed, since the objective function is quadratic and the constraints are linear, this is a quadratic program. 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 in equation (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 that 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 in equation* (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 define 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. No-trade is feasible.

3. Strict concavity: The objective V is strictly concave since the Hessian matrix, $-\mathbf{D}$, is negative definite.

It is a well-known principle of convex analysis that a strictly concave objective function on a non-empty compact and convex set has a unique maximizing vector $\mathbf{x}^*$ (Bertsekas (2009, Propositions 3.1.1, 3.2.1)).

□

Our approach makes the problem compact by assuming that traders are not interested in trading additional quantities beyond some very favorable 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 ($\mathbf{W}\mathbf{x} = \mathbf{0}$); (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 of market prices for assets, we use the notation $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)^\top$ for these multipliers. Two vectors of order-execution-rate multipliers, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_I)^\top$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_I)^\top$, 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 in equation (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 in equation (13) is formulated as an infimum rather than a 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 in equation* (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. Strict 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 market and is allowed on each order.[12]

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 as it does in our problem (Bertsekas (2009, Proposition 5.3.4)). 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 ($\mathbf{x} > \mathbf{q}$).

Theorem 2 does not guarantee that clearing prices are unique. The set of clearing prices is convex and may be unbounded. 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 buy order and one sell order for the same asset (or portfolio) with the same quantities q, and the buyer's lower limit price exceeds the absolute value of the seller's lower limit price. In this case, there is an interval of prices where both orders are fully executable. We discuss a tie-breaking rule to pick a unique price in the next section.

5 Computation

In this section, we study the computational feasibility of flow trading. The objective is to provide a proof of concept, finding market-clearing solutions in less than a second for a reasonably difficult problem with 500 assets and 100,000 orders, using an ordinary workstation. We also study how computation time varies with the number of assets and orders and parameters that affect how orders are generated. Section 5.1 proposes a com-

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

putational methodology. Section 5.2 explores computational performance in a simulation environment.

5.1 Methodology

Gradient Method For economists, Walrasian tatonnement is an intuitive approach for calculating market-clearing prices. An auctioneer announces tentative prices, and traders respond with their quantities. The auctioneer then adjusts prices in the direction proportional to net excess demand. The process continues until the market clears.

Tatonnement is equivalent to applying the gradient optimization method to an objective function of prices whose first-order conditions correspond to market clearing. In our setting, such a function can be obtained as

$$G(\boldsymbol{\pi}) := \inf_{\boldsymbol{\lambda},\boldsymbol{\mu}} \hat{G}(\boldsymbol{\pi},\boldsymbol{\lambda},\boldsymbol{\mu}) \qquad \text{subject to} \qquad \boldsymbol{\mu} \geq \mathbf{0}, \qquad \boldsymbol{\lambda} \geq 0, \tag{15}$$

where $\hat{G}(\boldsymbol{\pi},\boldsymbol{\lambda},\boldsymbol{\mu})$ is given by equation (12). Theorem 2 implies that this function's first order conditions, $G'(\boldsymbol{\pi}) = \mathbf{0}$, correspond to market clearing.

Since the gains function has a piecewise-linear derivative, it is continuously differentiable, and the derivative satisfies a Lipschitz condition.[13] These conditions assure that the gradient method converges (Nesterov (2004, Corollary 2.1.2, p. 70)). While the guaranteed convergence rate is much faster than for the traditional general-equilibrium theory problems discussed by Scarf and Hansen (1973),[14] it is too slow for our purpose. Reducing the error by a factor of one million may require approximately one million iterations, a prohibitively large number in our setting, where we need to solve for prices frequently throughout the trading day (as opposed to a single high-stakes allocation problem in a combinatorial auction).

5.1.1 Interior Point Method

We solve the minimization problem using an interior point method for quadratic programming. The literature shows that interior point methods are computationally more

[13] There is a Lipschitz constant L such that $|\nabla G(\boldsymbol{\pi}+\Delta\boldsymbol{\pi}) - \nabla G(\boldsymbol{\pi})| < L|\Delta\boldsymbol{\pi}|$ for all $\boldsymbol{\pi}$ and all $\Delta\boldsymbol{\pi}$.

[14] 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.

efficient than the more intuitive gradient method, both theoretically (see Nesterov (2004, Chapter 4); Bertsekas (2009), Boyd and Vandenberghe (2004)) and in practice (Gondzio (2012)).[15]

Exchange as a Small Market Maker Theoretically, interior point methods work better when the starting points are interior points, a feasible allocation on the interior of the constraint set. Such an allocation clears the market and strictly satisfies the inequality constraints ($\mathbf{0} < \mathbf{x} < \mathbf{q}$). In our setting, the natural candidate for an interior point is no-trade ($\mathbf{x} = \mathbf{0}$), which satisfies market clearing but is not an interior point because $\mathbf{x} = \mathbf{0}$ lies on the boundary, not the interior, of the feasible set. There is no other natural choice for an interior point.

To ensure an interior point, we let the exchange act as a small market maker for every asset. Specifically, the exchange submits a linear demand curve for each asset n,

$$\epsilon_n(\pi_{0n} - \pi_n), \tag{16}$$

where ϵ_n is the slope, and π_{0n} is a base price below which the exchange buys and above which it sells. Here, ϵ_n can be a small positive number such that the exchange does little trading. The strategy can be implemented by placing two flow orders for each asset: one order to buy at prices below π_{0n} and the other to sell at prices above π_{0n}, with a generous upper bound on the maximum quantity traded by the exchange.[16] With the exchange as a small market maker, existence of an interior point is assured. For example, pick any $\mathbf{x}$ such that $\mathbf{0} < \mathbf{x} < \mathbf{q}$. Then the exchange can soak up any uncleared quantities to clear the market.

Allowing modest exchange trading has two other benefits. First, it resolves the tiebreaker

[15] For interior point methods, the maximum number of iterations has an upper bound proportional to $O(\log(1/\epsilon))$, where ϵ is the proportion by which the error is reduced (Nesterov (2004) Theorem 3.1). For example, reducing error by proportion 0.000001 (one-millionth) is $O(\log(1,000,000)) \approx O(13.8)$. For gradient methods, the upper bound is proportional to $O(1/\epsilon)$ or $O(1/\epsilon^2)$ depending on the structure of the problem (Gondzio (2012)).

[16] The buy order can be implemented with an upper limit of $p_H = \pi_{0n}$, a lower limit p_L such that $(p_H - p_L)$ is a large positive number, a portfolio weight vector with weight 1 on asset n and weight 0 on all other assets, and a quantity parameter of $q = \epsilon_n(p_H - p_L)$. The sell order can be implemented analogously but with $p_H = -\pi_{0n}$. One very conservative way to set the lower limit price is to choose p_L such that $\epsilon_n(p_H - p_L)$ is the nth element of the matrix-vector product $\text{abs}(\mathbf{W})\mathbf{q}$, where $\text{abs}(\mathbf{W})$ is the element-by-element absolute value of the portfolio weight matrix $\mathbf{W}$. This guarantees that the exchange can, in principle, take the other side of any combined quantities of all other market participants (satisfying $\mathbf{0} < \mathbf{x} < \mathbf{q}$), even for extreme prices diverging towards plus or minus infinity.

problem, which arises when the convex set of market-clearing prices contains more than one point (as we know is otherwise possible from Theorem 2). Since the exchange has an active order for every asset at every potentially market-clearing price vector, market prices are chosen uniquely for all assets when multiple prices would otherwise be possible. For example, if π_{0n} is set at the previous clearing price for asset n, then the exchange's small trading demand will tend to break ties in favor of the prices closest to the previous prices. Second, the exchange can absorb uncleared quantities due to rounding error and inexact algorithm convergence to clearing prices, even when the algorithm has converged to a target tolerance.

Solving the KKT Conditions We use a primal-dual interior-point method to solve the Karush–Kuhn–Tucker (KKT) conditions. This approach finds market-clearing prices and quantities utilizing information about both quantities from the primal problem and prices and multipliers from the dual problem.

From here on, we redefine $\mathbf{p}^H$, $\mathbf{p}^L$, $\mathbf{D}$, $\mathbf{W}$, $\mathbf{q}$, and $\mathbf{x}$ to include the exchange's orders. Then the results from Section 4 hold, and it is straightforward to show that a solution to the KKT conditions clears the market. Further, since the exchange has an active order at any market clearing price, the solution is unique.

Theorem 3 (Karush–Kuhn–Tucker (KKT) Conditions with Exchange Trading)**.** *Any solution of the KKT conditions in equations* (17)–(20) *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:*

$$\mathbf{W}\mathbf{x}^* = \mathbf{0}, \qquad \mathbf{0} \le \mathbf{x}^* \le \mathbf{q}, \qquad \textit{(Primal Feasibility)} \tag{17}$$

$$\boldsymbol{\pi}^* \in \mathbb{R}^N, \qquad \boldsymbol{\lambda}^* \ge \mathbf{0}, \qquad \boldsymbol{\mu}^* \ge \mathbf{0}, \qquad \textit{(Dual Feasibility)} \tag{18}$$

$$\mathbf{p}^H - \mathbf{D}\mathbf{x}^* - \mathbf{W}^\top \boldsymbol{\pi}^* + \boldsymbol{\mu}^* - \boldsymbol{\lambda}^* = \mathbf{0}, \qquad \textit{(Primal Optimality)} \tag{19}$$

$$\boldsymbol{\lambda}^* \cdot (\mathbf{q} - \mathbf{x}^*) = \mathbf{0}, \qquad \boldsymbol{\mu}^* \cdot \mathbf{x}^* = \mathbf{0}, \qquad \textit{(Complementary Slackness)} \tag{20}$$

where the dot product in equation (20) *represents element-by-element multiplication of vectors. With exchange trading defined in equation* (16)*, there exists a unique solution to the KKT conditions.*

Proof of Theorem 3. Existence 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. Uniqueness follows from the upper bound on the quantity traded by the exchange being generous enough to insure that the exchange has a partially executable order for every asset at market clearing prices. If market-clearing prices were not unique, then any change in the price of any asset would change the aggregate quantity demanded, which implies multiple market-clearing quantities. Since the quantities are unique from Theorem 1, prices must therefore also be unique. □

Instead of solving these conditions directly, the interior point method first modifies the problem by replacing the complementary slackness conditions in equation (20) with a set of constraints parameterized by a scalar $\bar{\nu} > 0$:

$$\boldsymbol{\lambda}^* \cdot (\mathbf{q} - \mathbf{x}^*) = \bar{\nu}\,\mathbf{1}, \qquad \boldsymbol{\mu}^* \cdot \mathbf{x}^* = \bar{\nu}\,\mathbf{1}. \tag{21}$$

Then in the limit as $\bar{\nu} \to 0$, the sequence of solutions to the modified KKT conditions satisfies the original KKT conditions in Theorem 3.

The modified complementary slackness conditions in equation (21) imply that a solution to the modified KKT conditions satisfies the constraints with strict inequality: $\mathbf{0} < \mathbf{x} < \mathbf{q}$. Exchange trading plays a role in guaranteeing the existence of such a solution for any $\bar{\nu} > 0$.

Implementation Details Our algorithmic strategy solves the modified KKT conditions in equations (17), (18), (19), and (21) iteratively by starting with an initial guess for $\mathbf{x}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$ satisfying $\mathbf{0} < \mathbf{x} < \mathbf{q}$ (interior point), $\boldsymbol{\mu} > \mathbf{0}$, $\boldsymbol{\lambda} > \mathbf{0}$ (positive multipliers).[17] To find search directions ($\Delta\mathbf{x}$, $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\mu}$, and $\Delta\boldsymbol{\lambda}$), we first substitute $\mathbf{x} + \Delta\mathbf{x}$, $\boldsymbol{\pi} + \Delta\boldsymbol{\pi}$, $\boldsymbol{\mu} + \Delta\boldsymbol{\mu}$, and $\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$ for $\mathbf{x}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\lambda}$, respectively into the system of equations representing the modified KKT conditions with the value of $\bar{\nu}$ set to 0. Then we linearize the system of equations by dropping the second order terms ($\Delta\mathbf{x} \cdot \Delta\boldsymbol{\mu}$ and $\Delta\mathbf{x} \cdot \Delta\boldsymbol{\lambda}$ in the modified complementary

[17] Our own Python implementation of the interior point methodology follows the algorithm described by Vandenberghe (2010) for the CVXOPT package and is tailored to our specific quadratic program (which has an invertible diagonal matrix $\mathbf{D}$ and simple "Euclidean cone" constraints $\mathbf{0} \leq \mathbf{x} \leq \mathbf{q}$). One version of the algorithm is implemented on cpus using the Python packages numpy and scipy. Another version is implemented on both cpu and gpu using the Python package Pytorch. Results are reported for the Pytorch implementation on the gpu, which was three times faster than either cpu version. Our implementation code will be posted publicly upon publication and is available immediately to interested readers upon request. The Python programming language and the CVXOPT package (not actually used) are free and publicly available.

slackness conditions in equation (21)) and solve the resulting linear system for $\Delta\mathbf{x}$, $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\mu}$, and $\Delta\boldsymbol{\lambda}$.[18] The solution vectors are then multiplied by a scalar α (with $0 < \alpha \leq 1$) to ensure that the best guess for the next iteration $\mathbf{x}+\alpha\Delta\mathbf{x}$, $\boldsymbol{\pi}+\alpha\Delta\boldsymbol{\pi}$, $\boldsymbol{\mu}+\alpha\Delta\boldsymbol{\mu}$, $\boldsymbol{\lambda}+\alpha\Delta\boldsymbol{\lambda}$ is such that $\mathbf{x}$ remains an interior point and the multipliers remain strictly positive, with $\bar{v}$ eventually approaching zero. Since the KKT conditions are essentially first-order conditions, the linearized approximation is a version of Newton's method.

On each iteration, the linear system is solved in the following way. The multipliers $\Delta\boldsymbol{\mu}$ and $\Delta\boldsymbol{\lambda}$ are expressed as functions of $\Delta\mathbf{x}$, 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 $\boldsymbol{\pi}$, for which a Cholesky decomposition is used.[19] A mathematical derivation of the algorithmic details is in Appendix B.

We can think of the positive definite matrix to be decomposed as a "liquidity matrix" measuring the marginal change in quantities for each asset as a function of small changes in prices for all assets, taking into account both demand for individual assets and demand for portfolios. This liquidity matrix changes with each iteration because it is constructed by implicitly assigning weights to each order based on changing values of multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$. The weights are close to zero when the multipliers push the order execution rate x_i close to the boundary of the interval $[0, q_i]$, and closer to one if the execution rate x_i implied by the multipliers is closer to the midpoint of the interval $[0, q_i]$. The order is expected to be relevant for price discovery at the margin when it is partially executable, which, in the context of an interior point method, means that the weight implied by the multipliers is relatively closer to one than to zero. A new Cholesky decomposition is needed on each iteration to incorporate updated weights from the most recent iteration into the calculation of the new search direction.

When there is great liquidity for some portfolios (e.g., the market index) but little liquidity for some other portfolios (e.g., thinly traded individual assets), the matrix to be

[18] The KKT system is nonlinear in the unknowns $\boldsymbol{\pi}$, $\mathbf{x}$, $\boldsymbol{\mu}$, and $\boldsymbol{\lambda}$ only because the complementary slackness condition in equation (21) involves element-by-element multiplication of $\mathbf{x}$ by $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$. For $\boldsymbol{\mu}$ (and analogously for $\boldsymbol{\lambda}$), linearizing $(\mathbf{x}+\Delta\mathbf{x})\cdot(\boldsymbol{\mu}+\Delta\boldsymbol{\mu})$ sets the second-order term term $\Delta\mathbf{x}\cdot\Delta\boldsymbol{\mu}$ to a vector of zeros.

[19] As in the original KKT system, the revised KKT system is nonlinear because the revised complementary slackness condition involves element-by-element multiplication of $\mathbf{x}+\Delta\mathbf{x}$ by $\boldsymbol{\mu}+\Delta\boldsymbol{\mu}$ and $\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}$. To correct for the error created by dropping the second-order terms $\Delta\boldsymbol{\mu}\cdot\Delta\boldsymbol{\lambda}$, we solve the linear system a second time on each iteration (using the same Cholesky decomposition), including a correction term described by Mehrotra (1992) in the second solution.

decomposed is poorly conditioned and nearly singular. By supplying small amounts of liquidity to all assets, exchange trading improves the condition number of this matrix, which makes the Cholesky decomposition unlikely to fail. When there is insufficient exchange trading to prevent the Cholesky decomposition from failing, the algorithm regularizes the matrix by adding small quantities to the diagonal. While this makes the algorithm more robust, it may increase the number of iterations by making search directions less accurate.

5.2 Results

5.2.1 Simulating the Order Book

We simulate an order book with parameter settings designed to make the problem realistic and algorithmically difficult. Our goal is to provide a proof of concept, demonstrating that market clearing prices for 500 assets and 100,000 orders can be solved in less than one second. The number 500 is chosen based on the number of stocks in the S&P 500 index. The number 100,000 is chosen somewhat arbitrarily. After examining the base case, we show how the numbers of assets and orders affect computation times. We also test the robustness of the algorithm by examining how computation times vary when parameters values are much larger or smaller than base-case settings.

Of the 100,000 orders, 50,000 are for individual assets (an average of 100 orders for each asset), 25,000 are for various index portfolios, and 25,000 are for pairs trades. There are an additional 1000 exchange orders, one buy order and one sell order for each asset, making 101,000 orders altogether. The exchange offers a schedule with the tiny slope of 0.01. It is designed to do minimal trading, buying one dollar's worth of an asset when its price falls by one percent.

To generate a mix of assets – some very high volume some thinly traded, we create great variation in the number of orders across assets and the size of orders within assets. The expected number of orders for each asset and the size of orders for a given asset both follow lognormal distributions with large log-standard-deviations of 1.7 and 1.5, respectively.[20] This results in a few individual assets having a large number of orders and some assets having zero or very few orders. Within assets, a few orders are expected to be gigantic and most are expected to be relatively tiny. Mean order size is defined

[20] A log-standard-deviation of 1.7 means that a plus-or-minus one standard deviation change in the log of a random quantity multiplies or divides the random quantity by $e^{1.7} \approx 5.47$.

using the market microstructure invariance hypothesis of Kyle and Obizhaeva (2016), which makes mean order size for an individual asset or index proportional to the cube root of expected dollar volume for the individual asset or index.

While theoretically investors can choose from infinitely many different portfolios by combining any of the 500 assets, we restrict portfolios to various index portfolios and arbitrary pairs trades. For index portfolios, we construct value-weighted and equal-weighted portfolios of the market index, "size" indices, and "industry" indices. "Size" is defined as expected dollar volume.[21] "Industry" indexes are defined by arbitrarily grouping stocks so that the number of stocks and distribution of size do not vary across industries. Trading in indexes is dominated by the value-weighted market index. Pairs trades randomly buy either an asset or an index portfolio and sell an equal expected dollar value of another random asset or index.

Limit prices are distributed around an arbitrary initial price, normalized to \$100 per share or index unit. This is also the price at which the exchange trades a zero quantity.[22] For individual assets and index portfolios, the midpoint between upper and lower limit prices has a lognormal distribution centered at \$97.00 for buy orders and \$103.00 for sell orders, with an arbitrary log-standard-deviation of 10%. The expected difference between upper and lower prices is assumed to be very small, one basis point (i.e., $p_i^H = 100.005$, $p_i^L = 99.995$), with a very large log-variance of 2.0. Orders have equal probabilities of buying or selling the given individual asset or portfolio.

These assumptions are realistic because market indexes (like the S&P 500 E-mini futures contract or the SPDR ETF) have greater liquidity than any single stock, long-short trading is widely practiced, and liquidity and trading volume vary enormously across stocks. These assumptions also make the problem algorithmically difficult because huge variation in liquidity across assets and portfolios makes the liquidity matrix very poorly conditioned, and high-volume index trading volume makes the liquidity matrix highly non-diagonal. Further, huge variation in order size and the tiny difference between upper and lower limit prices stress the algorithm by making the liquidity matrix change a great deal when prices change by small amounts. The small difference between upper and lower limit prices is intentionally unrealistic since theory implies that market

[21] The simulation environment has no concept of market capitalization from which to define size. If it is assumed that all stocks have the same expected turnover rate of unmodeled market capitalization, then market capitalization is perfectly correlated with expected dollar volume.

[22] The simulations are structured so that normalizing prices scales quantities and prices but does not otherwise affect the algorithm (e.g., dollar values of all orders and trades are unaffected by this scaling).

Description	Base	Low	High
Number of assets	500	.	.
Number of orders	100000	.	.
Slope of exchange's demand schedule (shares traded per dollar price change at $100/share)	0.0100	.	.
Fraction of orders for individual assets	0.5000	0.0500	0.9500
Fraction of orders for indexes among orders for portfolios	0.5000	0.0500	0.9500
Number of size indexes	5	2	50
Number of industry indexes	10	2	50
Probability an index order is a market index order	0.8000	0.0500	0.9500
Probability a size or industry index order is a size index order	0.5000	0.0500	0.9500
Probability a mkt index order is an EW mkt index order	0.0625	0.0500	0.9500
Probability a size index order is an EW size index order	0.2500	0.0500	0.9500
Probability an industry index order is an EW industry index order	0.2500	0.0500	0.9500
Standard deviation of expected number of orders across assets	1.7000	0.1000	3.0000
Standard deviation of order size given asset	1.5000	0.1000	3.0000
Standard deviation of upper limit price as fraction of initial price	0.1000	0.0100	1.0000
Mean deviation of upper limit price as fraction of initial price standard deviation	0.3000	0.0100	1.0000
Mean difference between upper and lower limit prices (basis points)	1.0000	0.0100	100.0000
Standard deviation of difference between upper and lower limit prices	2.0000	0.1000	3.0000
Fraction buy orders for indexes and assets	0.5000	0.1000	0.9500

Table 2: Parameters for simulating an order book.

participants should supply more constant liquidity.

In addition to the base-case simulation parameters described above, we also test the robustness of the algorithm by using low and high values of parameters. Table 2 lists base-case, low, and high values for the parameters. Further details on the simulation methodology are provided in Appendix A.

5.2.2 Computation Outcomes

When performed on an ordinary office workstation—an AMD Ryzen Threadripper 3960X processor, 24 cores running at 3.8GHz, and 128GB of memory running at 3600MHz; RTZ 2070 gpu at 1710 MHz with 8 GB of RAM—computation of market-clearing prices and quantities takes about 0.1451 seconds (median) in the baseline scenario with 500 assets and 100,000 orders. Our results are obtained using the gpu and two cores.[23] Uncleared quantities are near zero, equal to 8.7 dollars per trillion dollars of total volume.[24]

The amount of exchange trading is small. On average, the exchange trades 3.2 dollars per million dollars of trading volume. Across 51 repetitions, the maximum, minimum,

[23] Computation times do not change much when more cores are used. This is probably because easily parallelized computations are done on the gpu, and other calculations do not benefit from using multiple cores. The computation times are stable across 401 repetitions, with a maximum of 0.1603 seconds, a minimum of 0.1365 seconds, and a standard deviation of 0.0058 seconds.

[24] All calculations are done with 64-bit floating-point numbers. If calculations are done with 32-bit floating-point numbers, computation time is more than twice as fast, but uncleared quantities are typically an unacceptably large $100 per million.

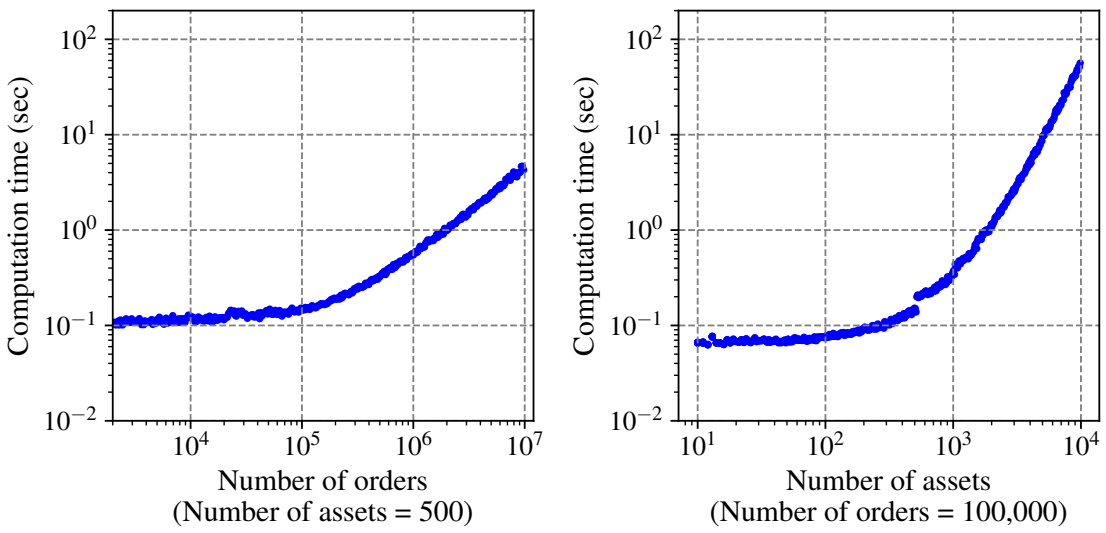

Figure 2: Computation Time As a Function of the Number of Orders and the Number of Assets

Panel A varies the number of orders. Panel B varies the number of assets. In both panels, all other parameters are set to their baseline values. Each dot represents one simulated order book, and there are approximately 500 simulations in each panel. The small discontinuity in Panel B around 600 assets likely is a hardware artifact, such as the need to use RAM rather than cache for sufficiently large problems.

and standard deviation of exchange trading are 5.99, 2.32, and 0.73 dollars per million dollars. In a dynamic market, the exchange can avoid accumulating significant inventories by adjusting its base prices over time to liquidate existing inventories.

Exchange trading, while small, has a significant effect on the cross-sectional standard deviation of market-clearing prices for assets with thin order books. Without exchange trading, assets with thin order books have essentially indeterminate prices. The algorithm chooses arbitrarily among multiple clearing prices. This drives the standard deviation of price changes from the baseline price to an immense value of 14.4 million percent. The small amount of exchange trading used in the simulations brings the standard deviation of prices down to a much more reasonable 14.19%, a reasonable magnitude similar to the 10% standard deviation of the midpoint limit price on orders for individual assets. In practice, we would expect market-making firms to provide liquidity in thinly traded stocks and stabilize prices, possibly with a wide spread.

Figure 2 describes how computation times vary with the number of assets and the number of orders. In the first panel, as the number of orders increases from 100,000 to 1,000,000 and 10,000,000, while keeping the number of assets constant, computation

times increase from 0.1451 to 0.5639 and 4.67 seconds. The computation time crosses one second with approximately 1,930,000 orders. When the number of orders is large, computation time is approximately proportional to the number of orders. In the second panel, as the number of assets increases from 500 to 2,000 and 10,000, again keeping the number of orders constant, computation times increase to 1.1021 and 56.3 seconds. The computation time crosses one second with 1,800 assets and ten seconds with 5,200 assets. The increased computation times when the number of assets increase are mainly due to the computation costs of constructing the liquidity matrix input to the Cholesky decomposition and the Cholesky decomposition itself (which is an $O(N^3)$ algorithm in the number of assets).[25]

For robustness, we alter each parameter's value to the minimum and the maximum of a wide range, as described in Table 2, while keeping the number of orders, the number of assets, and the slope of exchange trading constant. Computation times remain of the same order of magnitude (0.1159 to 0.2655 seconds compared to 0.1459 seconds in the baseline setting). Most parameters have a modest effect on computation times except for the standard deviation of order size and the fraction of buy orders. These two parameters affect the balance of the supply and demand of the order book. Changing the standard deviation of order size from 1.5 to 3 increases computation time to 0.2078 seconds. Changing the fraction of buy orders from 0.5 to 0.1 increases computation time to 0.2344 seconds. By making the order book more asymmetric, extreme values for these parameters make the problem more difficult to solve.

While having little effect on computation times, the difference between upper and lower limit prices significantly affects the quality of market clearing. The low mean difference scenario (0.01 basis points) and the high standard deviation scenario (3.00) increase the amount of uncleared quantities from 8.7 dollars per trillion to 120 and 160 dollars per trillion. Further making the mean difference ten times smaller makes uncleared quantities ten times larger. This occurs because as the upper and lower limit prices become closer ($p_i^H \to p_i^L$), the upper limit price also becomes closer to the market-clearing price ($p_i^H \to p_i$) for all executable orders. The ratio between the two differences $((p_i^H - p_i)/(p_i^H - p_i^L))$, each of which approaches zero, becomes numerically inaccurate. This makes the flow demand in equation (2), which depends on the ratio—and thus

[25] Both figures are almost flat initially. With a small number of orders or assets, the overhead associated with the Python interpreter becomes a significant fraction of computation times. When there are only 10 assets and 20 orders, the computation time is about 0.0552 seconds, which we believe is likely a good estimate of the overhead associated with the Python interpreter.

market-clearing quantities—inaccurate and increases uncleared quantities.

Finally, we consider an extreme scenario by setting all parameters simultaneously to those that increase computation times (either the minimum or the maximum of the range depending on the parameters). In this case, the computation time increases to 0.4291 seconds, approximately a factor of three relative to the base case and still below half a second, and the uncleared quantities increase to 1.5 dollars per million of total volume. The unclear quantities increase approximately by 100,000 fold relative to the baseline. This mainly results from reducing the mean and increasing the standard deviation of the differences between the upper and lower limit prices and increasing the standard deviation of the order size, each of which increases uncleared quantities by 100 fold by making the demand curve close to a step function and increasing the rounding errors from inferring quantities from prices. Still, uncleared quantities are a small fraction of the total volume, representing numerical errors from extreme parameter values as opposed to revealing economic issues (Theorems 1 and 2 show that market clearing prices and quantities exist). The analysis suggests that computation times are not sensitive to the parameter values used for the order book construction, while market clearing accuracy is sensitive to order book parameters, especially the difference between the two limit prices. While 1.5 dollars per million is still arguably small as an amount of inaccuracy, it may be prudent to adopt a lower bound for the difference between the upper and lower limit prices, such as one basis point in the baseline case.

Discussion Overall, market-clearing allocations are computed quickly and accurately with minimal trading by the exchange. Computational times grow with the number of orders and assets, as expected. Reassuringly, the growth in the number of orders appears to be linear, and problems with several thousand assets can be solved in about one to ten seconds on an ordinary workstation. We interpret these results as an initial computational proof of concept for the flow trading market design.

In a production environment, we expect more powerful computers and more refined algorithms will make it easier to calculate market-clearing allocations with even greater speed. Future algorithms may be better able to take advantage of parallel processing across more cores and take advantage of advances in quadratic programming and sparse matrix multiplications, which play key roles in our computation and are active areas of research in computer science.

6 Microfoundation for Portfolio Orders

Flow orders specify demand for a user-specified portfolio as a function of the price of that portfolio. Although such portfolio orders are more general than limit orders, this language is still restrictive. In general, a market participant's demands depend on the complete vector of asset prices, not just on the price of the portfolio. This section provides a microfoundation for our approach to expressing trading demands.

6.1 The Static CARA-Normal Framework

The CARA-normal model (Grossman (1976), Grossman and Stiglitz (1980), Admati (1985)), in which agents have constant absolute risk aversion (CARA) and asset returns are joint-normally distributed, is widely used in economics and finance. We use the CARA-normal model to study trading portfolio orders. The model is static, so there is no distinction between trading in quantities and flows. Models that study dynamic strategic trading in the CARA-normal environment have found that trading gradually over time is optimal to manage price impact (Vayanos (1999); Du and Zhu (2017); Kyle, Obizhaeva, and Wang (2018), Sannikov and Skrzypacz (2016)). While these models focus on the case of a single risky asset, we conjecture that the insights would carry over to the trade of portfolios.

Assume there are N risky assets and one safe asset, whose return is normalized to one. Assume there is a single trader who subjectively believes that the risky assets' payoffs, denoted by vector $\mathbf{v}$, are joint-normally distributed with mean $\mathbf{m}$ and variance-covariance matrix $\mathbf{\Sigma}$. The trader has CARA preferences with risk aversion parameter A. There are no wealth effects with CARA preferences, so the trader's wealth is set to zero for simplicity.

Initially, consider the trader's optimization problem given a fixed, known set of prices—let $\boldsymbol{\pi}$ denote the vector of prices for the N risky assets. Assume that the trader is a perfect competitor who cannot affect these prices with their trading; we will discuss the case where the trader has price impact shortly. The trader's portfolio optimization problem, given her beliefs, risk preferences, and prices, is given by:

$$\max_{\boldsymbol{\omega}} \mathrm{E}\Big[-\exp^{-A(\mathbf{v}-\boldsymbol{\pi})^{\top}\boldsymbol{\omega}}\Big], \tag{22}$$

Joint normality allows us to transform the above into the quadratic optimization

problem:

$$\max_{\boldsymbol{\omega}} \left[(\mathbf{m} - \boldsymbol{\pi})^\top \boldsymbol{\omega} - \tfrac{1}{2} A \boldsymbol{\omega}^\top \boldsymbol{\Sigma} \boldsymbol{\omega} \right]. \tag{23}$$

The first order condition implies that the optimal portfolio is given by:

$$\boldsymbol{\omega}^* = (A\boldsymbol{\Sigma})^{-1}(\mathbf{m} - \boldsymbol{\pi}). \tag{24}$$

Observe that the optimal demand for each asset depends on its covariance with the other assets, via the associated row of the inverse covariance matrix, and the entire vector $\mathbf{m} - \boldsymbol{\pi}$. Thus, as is well known, demand for each asset generally depends on the prices of all assets.

Implementing the Optimum with Portfolio Orders If the prices $\boldsymbol{\pi}$ are known and fixed, the trader can implement their optimum as defined in equation (24) with a single portfolio order with portfolio weights $\mathbf{w}_i$ and quantity parameter $Q_i^{\max}$ such that $Q_i^{\max} \mathbf{w}_i = \boldsymbol{\omega}^*$. This single portfolio order would specify pricing parameters such that it is fully executable at the known prices.

What if the trader does not know the asset prices? This might reflect the environment in which prices are rapidly fluctuating over time. Next, we show that traders can implement their optimum according to equation (24) with portfolio orders, without any knowledge of prices. To do this, we need to rotate the asset space such that independent portfolios span it.

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

$$\boldsymbol{\Sigma} = \mathbf{U} \boldsymbol{\Delta} \mathbf{U}^\top, \tag{25}$$

where $\mathbf{U}$ is an orthonormal matrix, and $\boldsymbol{\Delta}$ is a diagonal matrix with nonnegative elements. Let $K \leq N$ denote the rank of $\boldsymbol{\Sigma}$, let δ_i denote the ith nonzero diagonal entry of $\boldsymbol{\Delta}$, and let $\mathbf{u}_i$ denote the corresponding column of $\mathbf{U}$.[26] Then we have

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{K} \frac{1}{\delta_i} \mathbf{u}_i \mathbf{u}_i{}^\top. \tag{26}$$

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

Using this, we can express the optimal portfolio in equation (24) as

$$\boldsymbol{\omega}^* = \sum_{i=1}^{K} \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{A\delta_i} \right) \mathbf{u}_i, \tag{27}$$

which is a combination of demand schedules for portfolios. Here, $\mathbf{u}_1, \ldots, \mathbf{u}_K$ are portfolios of assets, whereas in equation (24) demand was expressed in terms of individual assets. Since the portfolios are independent of one another and there is no wealth effect, the optimal portfolio chooses the demand for each of them separately as if in a single-asset model.[27] That is, the optimal demand for the ith portfolio is given by

$$\frac{1}{A\delta_i}(\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}), \tag{28}$$

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 portfolio $\mathbf{u}_i$. Since the demand for each portfolio only depends on the portfolio's price, traders can achieve the optimal trade in equation (24) by utilizing K orders for portfolios where each order is a function of that portfolio's price.

Recall, we require orders' demands for portfolios to be downward sloping. Since the optimal demand for each portfolio in equation (27) is decreasing in the portfolio's price, the demand is indeed downward sloping.

The theorem below summarizes the results.

Theorem 4. *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 (24)) 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 (27)).*

Practical Implementation We can decompose the expected utility from the optimal portfolio into the contribution of each rotated asset. Substituting the optimal portfolio in equation (27) into equation (23), and some algebraic manipulations (see details in

[27] Observe that in equation (24), if the covariance matrix $\boldsymbol{\Sigma}$ is diagonal, then the demand coefficients on each of the individual assets are scalars, so the optimal portfolio can choose the demand for each asset separately as if in a single-asset model too.

Appendix A), allows us to express the expected utility from trading at prices $\boldsymbol{\pi}$ as

$$\sum_{i=1}^{K} \frac{1}{2A} \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{\sqrt{\delta_i}} \right)^2. \tag{29}$$

This formula shows that the benefit of each portfolio is determined by its squared Sharpe ratio as perceived by the trader.[28] 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 K portfolios.

Price Impact and Strategic Trading Thus far, we have assumed that traders are perfect competitors, behaving as if they have no price impact. In practice, trades can move prices. Many institutional traders dedicate considerable time and resources to managing their price impact. Now we show that flow orders can still be used to implement the optimal portfolio when traders behave strategically, considering 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.[29] That is, for each trader, there is an $N \times N$ positive semidefinite matrix $\boldsymbol{\Lambda}$, such that

$$\boldsymbol{\pi} = \boldsymbol{\pi}_0 + \boldsymbol{\Lambda}\, \boldsymbol{\omega}, \tag{30}$$

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 $\boldsymbol{\Lambda}$ 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 (24), given by

$$\boldsymbol{\omega}^* = \left(A\boldsymbol{\Sigma} + \boldsymbol{\Lambda}\right)^{-1} \left(\mathbf{m} - \boldsymbol{\pi}\right). \tag{31}$$

[28] 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.

[29] 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.

Since the sum of two positive semidefinite matrices is also positive semidefinite, $A\boldsymbol{\Sigma}+\boldsymbol{\Lambda}$ is positive semidefinite. Thus, we can use singular value decomposition to rotate the asset space such that independent portfolios span it. 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 $A\boldsymbol{\Sigma}+\boldsymbol{\Lambda}$.

Theorem 5. *Consider a static CARA-normal framework in which a trader believes that her price impact is linear and positive semidefinite (equation* (30)*). Then the strategic trader's optimal portfolio (equation* (31)*) 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 4, we treat orders as if they represent traders' true valuations. This simplification does not imply that we can infer traders' valuations from their orders. Strategic trading is an important reason that there often is a gap between true and as-bid valuations.

6.2 Approximations for General Preferences and Limitations

The logic above extends to any strictly concave, twice continuously differentiable, quasilinear utility function over assets provided asset payoffs are joint normally distributed. To see why this is the case, recall the two CARA preference properties we use: 1) no wealth effects and 2) strict concavity.

First, with no wealth effects, a trader's optimal demand for each asset does not depend on the prices of other assets in the case where assets' payoffs are independent of one another. If the assets have correlated, joint normal payoffs, we can, as shown above, rotate the asset space such that it is spanned by a set of portfolios whose payoffs are independent of one another. Thus, any quasilinear preference implies no wealth effects, and we can use the independent portfolios such that the trader's optimal demand for each portfolio only depends on that portfolio's price.

Second, strict concavity implies that the optimal demand for any portfolio must be downward sloping. This is crucial since we require demands for portfolios to be downward sloping in the portfolio's price. Although the optimal strategy may not be linear for any strictly concave, twice continuously differentiable, quasilinear preference, we

can approximate the optimal strategy by combining multiple linear downward sloping schedules.

However, portfolio orders will be unable to approximate the optimal portfolio of every concave utility function. First, with wealth effects, the demand for an independent portfolio may still depend on the prices of other portfolios and may also increase in that portfolio's price. Second, with asymmetric information, the prices of other portfolios may be useful to learn about the payoff of a given portfolio, even if the payoffs of the two portfolios are independently distributed. In this case, the optimal demand for an independent portfolio may again depend on the prices of other portfolios.

7 Implementation and Policy Issues

This section discusses several practical implementation and policy issues related to the flow trading market design.

Batch Interval What is the optimal batch interval? The best choice likely depends on the asset class. We discuss three considerations: computation limits, factors that favor a shorter interval subject to computation constraints, and an open question about whether a longer interval is desirable for thinly traded assets.

The batch interval must be long enough to compute prices and trades. Our simulations suggest that a batch interval on the order of one second may be sufficient for many asset classes. That said, the computational simulations serve as a proof-of-concept rather than as a final word. There are many reasons why a real-world implementation could be meaningfully faster than our implementation. There is also the possibility that real-world markets may take longer to compute for reasons not anticipated by our simulation environment.

Next, three factors favor a batch interval as short as computationally feasible. First, a fast batch interval makes trading smoother—that is, smaller quantities are traded per batch. Smoother trading can be helpful to traders with complex dynamic trading strategies, who may wish to adjust their orders over time as information evolves. At the same time, traders with simpler strategies can leave their orders be, without much adjustment over time, whether the batch interval is long or short. Second, if the assets in question are traded in fragmented markets, and some of those markets are continuous, then the interaction of a discrete-time market with a continuous-time market is likely simplest

if the discrete-time interval is short (Budish, Lee, and Shim (2018)). Third, information policy is more robust with a shorter batch interval, as there will be less pressure for within-batch information dissemination.

Last, we acknowledge the open question of whether a longer batch interval is desirable for more thinly traded assets. This is a common intuition among regulators and practitioners (see, e.g., U.S. Securities and Exchange Commission (2019*a*,*b*), Schwartz (2012), and references therein). Du and Zhu (2017) provide some theoretical grounding for this intuition. A hard conceptual question is how to think about the batch interval for markets consisting of heavily traded and thinly traded assets. For instance, the US equities market consists of about 7000 assets, some of which trade many times per second while others trade only a few times per hour. Similarly, in many sovereign debt markets, on-the-run assets are heavily traded while off-the-run assets are thinly traded.

Information Policy Information policy is typically discussed in terms of pre-trade 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 aggregate net demand curve for each asset, holding the prices of all other assets fixed. This policy is analogous to publishing the outstanding limit order book in continuous markets. Then traders can make inferences about the price impact costs of their orders. The exchange would not publish information about the identity of traders, nor would it publish individual orders, since portfolio weights may reveal trading strategies.

For pre-trade transparency, we envision that the post-trade information from the auction at time t is the complete pre-trade information for the auction at time $t+1$. As discussed by Budish, Cramton, and Shim (2015), this is the appropriate discrete-time analog of information policy in the continuous market. In both cases, the exchange (i) receives an order, (ii) economically processes the order, and then (iii) disseminates information about what happened (e.g., a trade or an order book update). The difference with discrete time is that the economic processing in (ii) occurs in discrete time, and hence the information dissemination in (iii) occurs in discrete time as well.[30]

[30] As pointed out by Budish, Cramton, and Shim (2015), in the continuous-time market it may look like traders can see information about the state of an asset's order book "right now," but that is an illusion—a trader's information is always as of a latency ago because it takes non-zero time for the exchange's matching engine to economically process new messages in step (ii) and to disseminate updates in step (iii).

The reason not to disseminate additional information between batch auctions, e.g., about the arrival of new orders or the cancelations of outstanding orders, is that such information could lead to gaming. For example, suppose the batch interval is one second. A trader could submit a new order to buy a large quantity 100 milliseconds into the batch interval with the intention to cancel that order and instead send a new order to sell 999 milliseconds into the batch interval. In this scenario, the order to buy was never economically binding nor economically processed by the exchange, so sending this purposefully misleading message was costless.

We acknowledge that the pressure to disseminate information between auctions grows with the batch interval. This is one reason why a batch interval that is as short as computationally feasible may be appropriate for many asset classes. Additionally, one could extend flow trading to include order types that are economically binding for the duration of the current auction (i.e., cannot be canceled until after the next auction), with within-auction information disseminated about updates to this binding subset of the order book. Professional market-making firms might deploy such orders to attract trading volume, but we view this discussion as speculative and in need of future research.

With arbitrary portfolio orders, information about the depth of the order book is inherently complex—there are infinitely many possible portfolios. The exchange might publish limited depth information about a fixed list of reference portfolios, alongside the depth information for individual assets.

Trust and Transparency Flow trading has the desirable property that all orders that are executable at published clearing prices do in fact execute. This property allows investors to confirm that their orders received correct execution from published prices.

By contrast, potentially executable orders in current markets do not consistently execute when other orders execute at the same price at nearly the same time. Uncertain execution erodes trust and market confidence, particularly among traders without state-of-the-art speed technology, whose orders are more apt to get poorer execution.

This difference between flow trading and the current market design arises from combining discrete time and continuous prices and quantities. Continuous prices and quantities make it possible to execute all executable orders at a market-clearing price without

Discrete time makes more transparent that a trader's information as of time t is the state of the order book as of time $t-\Delta$, whether Δ is the latency of information travel in a continuous market or the duration of the batch interval in a discrete market. Discrete time also eliminates the arms race for speed to reduce Δ.

any need for rationing. Discrete time makes it possible to process multiple executable orders simultaneously in a batch process.

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. Slower traders perceive this as 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 perceive 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 each asset is continuous and strictly downward sloping. All trades clear at the same price. All executable orders execute. There are of course still trading costs. Trading a larger quantity, or trading a given quantity faster, requires offering a better price—that is, walking up the market's supply curve if buying or down the market's demand curve if selling—which creates price impact. The essential difference is that a trader can trade an epsilon quantity at the market-clearing price without any bid-ask spread. A practical interpretation of this point is that institutional investors will have to manage their price impact, but small retail investors can trade small quantities at the market-clearing price with negligible trading costs.

Regulatory Objectives The US Securities and Exchange Commission, which regulates US securities markets, pursues various policy objectives, including economic efficiency, competition, maintaining trust and confidence, and investor protection.

Flow trading is consistent with these objectives. It improves 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 reducing the messages needed to implement dynamic trading strategies. It increases competition by enabling institutional investors to trade arbitrary long-short portfolios without the need for sophisticated trading platforms. Flow trading is also 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 promotes trust and confidence in markets by trading all executable orders at the same transparent price. It protects investors from poor order execution by making quality of order execution easy for customers to measure.

8 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 and uniqueness of equilibrium prices and quantities and their fast computation. This seems a useful new point on the frontier of language design, that is, an attractive tradeoff between expressiveness and computability.

An open topic left for future research is the efficiency and welfare consequences of allowing market participants to directly trade portfolios, rather than trading separately for individual assets. We conjecture there are two main efficiency benefits. First, allowing market participants to directly trade portfolios should reduce trading costs, such as

the costs of trade execution and intermediation. This includes costs paid indirectly for portfolio trade execution through ETF fees. 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. This should improve the efficiency of price discovery and reduce arbitrage rents. Another interesting dimension of the problem is how allowing participants to trade portfolios affects strategic issues around demand- and supply-reduction to manage price impact. In models with fully-contingent trading demands, these effects can go in either direction (Rostek and Yoon (2021), Wittwer (2021)).

References

Admati, Anat R. 1985. "A noisy rational expectations equilibrium for multi-asset securities markets." *Econometrica*, 629–657.

Antill, Samuel, and Darrell Duffie. 2020. "Augmenting markets with mechanisms." *Review of Economic Studies*, 88(4): 1665–1719.

Aquilina, Matteo, Eric Budish, and Peter O'Neill. 2022. "Quantifying the high-frequency trading "arms race"." *Quarterly Journal of Economics*, 137(1): 493–564.

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 largesscale 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.

Budish, Eric, Robin Lee, and John Shim. 2018. "Will the market fix the market? A theory of stock exchange competition and innovation." University of Chicago Working Paper.

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.

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

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

Grossman, Sanford. 1976. "On the efficiency of competitive stock markets where trades have diverse information." *Journal of Finance*, 31(2): 573–585.

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.

Hasbrouck, Joel, and Gideon Saar. 2013. "Low-latency trading." *Journal of Financial Markets*, 16(4): 646–679.

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, and Alexander Westkamp. 2021. "Stability, strategy-proofness, and cumulative offer mechanisms." *Review of Economic Studies*, 88(3): 1457–1502.

Kelso Jr., Alexander S., and Vincent P. Crawford. 1982. "Job matching, coalition formation, and gross substitutes." *Econometrica*, 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.

Kyle, Albert S. 1989. "Informed speculation with imperfect competition." *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*.

MacKenzie, Donald. 2021. *Trading at the speed of light.* Princeton University Press.

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.

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. "Equilibrium theory of financial markets: Recent developments." *Journal of Economic Literature*.

Rostek, Marzena, and Ji Hee Yoon. 2021. "Exchange design and efficiency." *Econometrica*, 89(6): 2887–2928.

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.

Sannikov, Yuliy, and Andrzej Skrzypacz. 2016. "Dynamic trading: Price inertia and front-running." Working paper.

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

Schwartz, Robert A. 2012. *The electronic call auction: Market mechanism and trading: Building a better stock market.* Vol. 7, Springer Science & Business Media.

Shapley, Lloyd S, and Martin Shubik. 1971. "The assignment game I: The core." *International Journal of Game Theory*, 1(1): 111–130.

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

U.S. Securities and Exchange Commission. 2019*a*. "Commission statement on market structure innovation for thinly traded securities, Release No. 34-87327." Retrieved February 11, 2022 from https://www.sec.gov/rules/policy/2019/34-87327.pdf.

U.S. Securities and Exchange Commission. 2019*b*. "Division of trading and markets: Background paper on the market structure for thinly traded securities." Retrieved February 11, 2022 from https://www.sec.gov/rules/policy/2019/thinly-traded-securities-tm-background-paper.pdf.

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." *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.

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.

Appendix

A Simulation Details

In the base case, orders for index portfolios are randomly assigned to the six categories with corresponding probabilities in parentheses: the valued-weighted market index (75%), the equal-weighted market index (5%), five value-weighted size indices (7.5%), five equally-weighted size indices (2.5%), ten value-weighted industry indices (7.5%), and ten equal-weighted industry indices (2.5%). The numbers here are chosen somewhat arbitrarily. We later vary the probabilities to study how they may affect computation times.[31] Finally, each order for individual assets and indexes has an equal probability of being buy or sell.

For each asset, we draw a random number from a lognormal distribution with mean of 1 and log-standard deviation of 1.7. Dividing these numbers by the sum of all realizations across 500 assets, we generate the probability that a given order is allocated to that asset. Then for each order for individual assets, we pick an asset from a multinomial distribution with the chosen probabilities. The probability multiplied by the total number of orders for assets (50,000) is the expected number of orders for that asset.

Following the market microstructure invariance hypothesis of Kyle and Obizhaeva (2016), the mean order size is set proportionally to the square root of the expected number of orders for that asset. The proportionality constant is chosen to make the aggregate expected order volume from individual stocks equal to the arbitrary scaling constant of \$10 million per batch using arbitrary expected ex-ante prices of \$100 per share. Then the standard deviation of the order size equals $\sqrt{\exp(1.5^2)-1}$ multiplied by the mean, approximately two times the mean.

For index portfolios, the expected number of orders for each size index is the same, and the expected number of orders for each industry index is the same. The size of the index orders is determined by multiplying the square root of the expected number of

[31] To allow conveniently varying these probabilities, we generate them from five parameters: the probability that an index order is for either the equal-weighted or the value-weighted market index; the probability that a non-market index order is for a size index portfolio; the probability that a market index order is for the equal-weighted market index portfolio; and the probability that a size (industry) index order is for an equal-weighted size (industry) index portfolio. The five parameters and the restriction that the probabilities sum to one determine all six probabilities. We let each of the five parameters vary from 5% to 95%.

orders by the same proportionality factor used for individual orders. Since orders for the value-weighted market index are much larger and more numerous than orders for individual stocks, the overall value of the market index is primarily determined by these index orders. For pairs trades, each individual asset leg is generated randomly in the same manner as orders for the asset or portfolio. The dollar size of the larger leg is then truncated to match the dollar size of the smaller leg, again using expected ex-ante prices.

B Solving KKT Conditions

The system of equations representing the modified KKT conditions (19), and (17), (21) is (18), can be rearranged and written[32]

$$\mathbf{p}^H - \mathbf{D}\mathbf{x} - \mathbf{W}^\top \boldsymbol{\pi} + \boldsymbol{\mu} - \boldsymbol{\lambda} = \mathbf{0}, \tag{32}$$

$$\mathbf{W}\mathbf{x} = \mathbf{0}, \tag{33}$$

$$\boldsymbol{\lambda} \cdot (\mathbf{q} - \mathbf{x}) = \bar{\nu}\,\mathbf{1}, \qquad \boldsymbol{\mu} \cdot \mathbf{x} = \bar{\nu}\,\mathbf{1} \tag{34}$$

$$\boldsymbol{\lambda} \geq \mathbf{0}, \qquad \boldsymbol{\mu} \geq \mathbf{0}, \qquad \mathbf{0} \leq \mathbf{x} \leq \mathbf{q}, \tag{35}$$

In this system, the exogenous order book is represented by the matrix of portfolio weights $\mathbf{W}$, the vector of maximum portfolio quantities $\mathbf{q}$, the vector of upper limit prices $\mathbf{p}^H$, and the diagonal matrix $\mathbf{D}$.[33] The goal is to find values for the traded portfolio quantities $\mathbf{x}$, prices for assets (multipiers for market clearing constraints) $\boldsymbol{\pi}$, and multipliers for order quantity constraints $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ which solve this system for a very small positive value of the interior point parameter $\bar{\nu}$, which is given by

$$\bar{\nu} := \tfrac{1}{2m} \left(\boldsymbol{\lambda}^\top (\mathbf{q} - \mathbf{x}) + \boldsymbol{\mu}^\top \mathbf{x} \right). \tag{36}$$

The algorithm starts with a large initial guess for $\bar{\nu}$ and an initial guesses for $\mathbf{x}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\lambda}$, then calculates revised guesses iteratively, preserving the interior-point and nonneg-

[32] The actual algorithm used in the simulations replaces $\mathbf{q}-\mathbf{x}$ and $\mathbf{x}$ in the complementary slackness conditions with slack variable $\mathbf{s}_\mu$ and $\mathbf{s}_\lambda$, writes the approximation to the complementary slackness condition as $\boldsymbol{\mu} \cdot \mathbf{s}_\mu = \boldsymbol{\lambda} \cdot \mathbf{s}_\lambda = \bar{\nu} \cdot \mathbf{1}$, then solves for the slack variable along with the other variables. The slack variables quickly converge to their correct values $\mathbf{s}_\mu := \mathbf{x}$ and $s_\lambda := \mathbf{q} - \mathbf{x}$ and will always attain their correct values if initialized correctly. This approach is essentially equivalent to the slightly simplified exposition given here.

[33] The lower limit prices $\mathbf{p}^L$ are easily obtained from $\mathbf{p}^H$, $\mathbf{q}$, and $\mathbf{D}$ but are not needed for calculations.

ativity constraints (35). On a given iteration, the problem is linearized by substituting $\mathbf{x}+\Delta\mathbf{x}$, $\boldsymbol{\pi}+\Delta\boldsymbol{\pi}$, $\boldsymbol{\mu}+\Delta\boldsymbol{\mu}$, and $\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}$ for $\mathbf{x}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\lambda}$, respectively into this system of equations. This results in a system of equations which is linear in $\Delta\mathbf{x}$, $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\mu}$, and $\Delta\boldsymbol{\lambda}$, except for the non-linear terms $\Delta\boldsymbol{\mu}\cdot\Delta\mathbf{x}$ and $\Delta\boldsymbol{\lambda}\cdot\Delta\mathbf{x}$. Since these quadratic terms are unknown, they are replaced with guesses $\boldsymbol{\epsilon}_{\Delta\boldsymbol{\mu}\cdot\Delta\mathbf{x}}$ and $\boldsymbol{\epsilon}_{\Delta\boldsymbol{\lambda}\cdot\Delta\mathbf{x}}$, whose values are discussed in the paragraph after equation (48). Since the goal is to solve the system for small and smaller versions of $\bar{\nu}$, the value $\bar{\nu}$ is replaced by a smaller quantity $\epsilon_{\bar{\nu}}$. The theory of interior point methods is based on reducing $\bar{\nu}$ gradually iteration by iteration. In practice, the algorithm converges faster if large reductions are attempted. Here we set $\epsilon_{\bar{\nu}}=0$ to try to reduce $\bar{\nu}$ substantially on each iteration.

Placing terms linear in $\Delta\mathbf{x}$, $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\mu}$, and $\Delta\boldsymbol{\lambda}$ on the left side of equations, the linearized system can be written

$$\mathbf{D}\Delta\mathbf{x}-\mathbf{W}^{\top}\Delta\boldsymbol{\pi}+\Delta\boldsymbol{\mu}-\Delta\boldsymbol{\lambda}=-\mathbf{r}_x,\qquad\text{where}\qquad\mathbf{r}_x:=\mathbf{p}^H-\mathbf{D}\mathbf{x}-\mathbf{W}^{\top}\boldsymbol{\pi}+\boldsymbol{\mu}-\boldsymbol{\lambda},\tag{37}$$

$$\mathbf{W}\,\Delta\mathbf{x}=-\mathbf{r}_{\pi},\qquad\text{where}\qquad\mathbf{r}_{\pi}:=\mathbf{W}\mathbf{x}\tag{38}$$

$$\mathbf{x}\cdot\Delta\boldsymbol{\mu}+\boldsymbol{\mu}\cdot\Delta\mathbf{x}=-\mathbf{r}_{\mu}\qquad\text{where}\qquad\mathbf{r}_{\mu}:=\boldsymbol{\mu}\cdot\mathbf{x}+\boldsymbol{\epsilon}_{\Delta\boldsymbol{\mu}\cdot\Delta\mathbf{x}}-\epsilon_{\bar{\nu}}\mathbf{1},\tag{39}$$

$$(\mathbf{q}-\mathbf{x})\cdot\Delta\boldsymbol{\lambda}-\boldsymbol{\lambda}\cdot\Delta\mathbf{x}=-\mathbf{r}_{\lambda},\qquad\text{where}\qquad\mathbf{r}_{\lambda}:=\boldsymbol{\lambda}\cdot(\mathbf{q}-\mathbf{x})+\boldsymbol{\epsilon}_{\Delta\boldsymbol{\lambda}\cdot\Delta\mathbf{x}}-\epsilon_{\bar{\nu}}\mathbf{1}.\tag{40}$$

Now define some notation. For any vector $\mathbf{z}$, let $\mathbf{z}^{-1}$ denote the vector of element-by-element reciprocals of elements of $\mathbf{z}$. For any matrix $\mathbf{Z}$, let $\mathrm{diagvec}(\mathbf{Z})$ denote the vector on its diagonal. For any vector $\mathbf{z}$, let $\mathrm{diagmat}(\mathbf{z})$ denote the diagonal matrix with $\mathbf{z}$ on its diagonal. Note that for a diagonal matrix $\mathbf{Z}$, we can write the matrix-vector product as an element-by-element vector-vector product: If $\mathbf{z}:=\mathrm{diagvec}(\mathbf{Z})$, then $\mathbf{Z}\mathbf{v}=\mathbf{z}\cdot\mathbf{v}$.

Solve equations (39) and (40) for $\Delta\boldsymbol{\mu}$ and $\Delta\boldsymbol{\lambda}$:

$$\Delta\boldsymbol{\mu}=\mathbf{x}^{-1}\cdot(-\mathbf{r}_{\mu}-\boldsymbol{\mu}\cdot\Delta\mathbf{x})\tag{41}$$

$$\Delta\boldsymbol{\lambda}=(\mathbf{q}-\mathbf{x})^{-1}\cdot(-\mathbf{r}_{\lambda}+\boldsymbol{\lambda}\cdot\Delta\mathbf{x}).\tag{42}$$

Plug these solutions for $\Delta\boldsymbol{\mu}$ and $\Delta\boldsymbol{\lambda}$ into equation (37), putting terms linear in $\Delta\boldsymbol{\mu}$ and $\Delta\boldsymbol{\lambda}$ on the left side:

$$\mathbf{D}\Delta\mathbf{x}-\mathbf{W}^{\top}\Delta\boldsymbol{\pi}+\mathbf{x}^{-1}\cdot(-\mathbf{r}_{\mu}-\boldsymbol{\mu}\cdot\Delta\mathbf{x})-(\mathbf{q}-\mathbf{x})^{-1}\cdot(-\mathbf{r}_{\lambda}-\boldsymbol{\lambda}\cdot\Delta\mathbf{x})=-\mathbf{r}_x.\tag{43}$$

Define

$$\mathbf{d} = \text{diagmat}(\mathbf{D}), \qquad \boldsymbol{\omega} := \left(\mathbf{d} + \mathbf{x}^{-1} \cdot \boldsymbol{\mu} + (\mathbf{q} - \mathbf{x})^{-1} \cdot \boldsymbol{\lambda}\right)^{-1}, \qquad \boldsymbol{\Omega} = \text{diagmat}(\boldsymbol{\omega}). \tag{44}$$

Solve equation (43) for $\Delta\mathbf{x}$ to obtain

$$\Delta\mathbf{x} = \boldsymbol{\omega} \cdot \left(\mathbf{W}^\top \Delta\boldsymbol{\pi} - \mathbf{r}\right), \qquad \text{where} \quad \mathbf{r} := \mathbf{r}_x + \mathbf{x}^{-1} \cdot \mathbf{r}_\mu + (\mathbf{q} - \mathbf{x})^{-1} \cdot \mathbf{r}_\lambda + \mathbf{x} \cdot \mathbf{r}_\mu. \tag{45}$$

Define the "liquidity matrix" $\mathbf{L}$ as

$$\mathbf{L} = \mathbf{W}\boldsymbol{\Omega}\mathbf{W}^\top. \tag{46}$$

While the liquidity matrix $\mathbf{L}$ is theoretically positive definite (due to the exchange trading every asset), it may be numerically singular (due to tiny exchange trading). To regularize this matrix, add a vector of small positive values, denoted $\boldsymbol{\epsilon}_\mathbf{L}$, to the diagonal. Substitute this solution for $\Delta\mathbf{x}$ into the market clearing condition (38) ($\mathbf{W}\,\Delta\mathbf{x} = -\mathbf{r}_\pi$) to obtain

$$\left(\mathbf{L} + \text{diagmat}(\boldsymbol{\epsilon}_\mathbf{L})\right)\, \Delta\boldsymbol{\pi} = -\mathbf{r}_\pi + \mathbf{W}\boldsymbol{\omega} - \mathbf{r}. \tag{47}$$

Since the regularized liquidity matrix $\mathbf{L} + \text{diagmat}(\boldsymbol{\epsilon}_\mathbf{L})$ is positive definite and presumably not numerically singular, the above equation can be solved for $\boldsymbol{\pi}$ using a Cholesky decomposition. Solutions for $\Delta\mathbf{x}$, $\Delta\boldsymbol{\mu}$, and $\Delta\boldsymbol{\lambda}$ can be obtained from the previous equations.

Now these solutions may not be such that the updated vectors $\mathbf{x} + \Delta\mathbf{x}$, $\boldsymbol{\pi} + \Delta\boldsymbol{\pi}$, $\boldsymbol{\mu} + \Delta\boldsymbol{\mu}$, $\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$ satisfy the constraints (35) requiring that $\mathbf{x}$ be an interior point and multipliers be positive. To insure that the constraints hold, truncate the solutions by a factor $\bar{\alpha}$ defined by

$$\bar{\alpha} := 0.99 \sup\left[\alpha : 0 \le \alpha \le 1, \mathbf{0} \le \mathbf{x} + \alpha\Delta\mathbf{x} \le \mathbf{q}, \boldsymbol{\mu} + \alpha\Delta\boldsymbol{\mu} \ge \mathbf{0}, \boldsymbol{\lambda} + \alpha\Delta\boldsymbol{\lambda} \ge \mathbf{0}\right], \tag{48}$$

The factor 0.99 insures that the updated solutions $\mathbf{x} + \bar{\alpha}\Delta\mathbf{x}$, $\boldsymbol{\pi} + \bar{\alpha}\Delta\boldsymbol{\pi}$, $\boldsymbol{\mu} + \bar{\alpha}\Delta\boldsymbol{\mu}$, and $\boldsymbol{\lambda} + \bar{\alpha}\Delta\boldsymbol{\lambda}$ satisfy inequality constraints as strict inequalities.

Now consider how to choose the guesses $\boldsymbol{\epsilon}_{\Delta\boldsymbol{\mu}\cdot\Delta\mathbf{x}}$ and $\boldsymbol{\epsilon}_{\Delta\boldsymbol{\lambda}\cdot\Delta\mathbf{x}}$. On each iteration, the solution for $\Delta\mathbf{x}$, $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\mu}$, and $\Delta\boldsymbol{\lambda}$ is calculated twice reusing the same Cholesky decomposition. On the first try, the guesses are $\boldsymbol{\epsilon}_{\Delta\boldsymbol{\mu}\cdot\Delta\mathbf{x}} = \boldsymbol{\epsilon}_{\Delta\boldsymbol{\lambda}\cdot\Delta\mathbf{x}} = \mathbf{0}$. On the second try, the solution is polished using the results from the first try as guesses (Mehrotra (1992)):

$$\boldsymbol{\epsilon}_{\Delta\boldsymbol{\mu}\cdot\Delta\mathbf{x}} = \bar{\alpha}^2 \Delta\boldsymbol{\mu} \cdot \Delta\mathbf{x}, \qquad \boldsymbol{\epsilon}_{\Delta\boldsymbol{\lambda}\cdot\Delta\mathbf{x}} = \bar{\alpha}^2 \Delta\boldsymbol{\lambda} \cdot \Delta\mathbf{x}. \tag{49}$$

The initial guess for $\mathbf{x}$ is rather arbitrary, involving large values for the multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$.

Computationally, calculation of the matrix $\mathbf{L} = \mathbf{W}\boldsymbol{\Omega}\mathbf{W}^\top$ and its Cholesky decomposition are the most costly parts of the algorithm. The next most costly calculations are several matrix-vector products involving the sparse portfolio weight matrix $\mathbf{W}$. The remaining calculations are relatively less costly element-by-element vector products, scalar products, and inner products.

To make calculations involving $\mathbf{W}$ more computationally efficient, the matrix $\mathbf{W}$ is expressed as the product of two matrices. The first matrix is a vector of weights defining portfolios. It concatenates an identity matrix (defining "weights" for individual assets) with another matrix whose columns define weights for index portfolios. The second matrix has one column for each order, with one non-zero weight defining the individual asset or portfolio traded by orders for individual assets or portfolios and with two non-zero weights for the two portfolios involved in pairs trades. Since both of these matrices are sparse, there is computational savings from not forming the matrix $\mathbf{W}$ explicitly but instead performing matrix multiplications involving $\mathbf{W}$ in a "matrix-free" manner by multiplying by the two matrices sequentially. We use an ad hoc algorithm exploiting the specific structure of these matrices. For example, with 20,000 orders for the market portfolio of 500 assets in the base-case scenario, the entire vector of 500 portfolio weights defining the market portfolio is only multiplied once rather than 20,000 times, as would be the case if the matrix $\mathbf{W}$ were calculated explicitly. Measuring the effect of this approach on computational efficiency in different order books with more complicated kinds of portfolio orders is an interesting area for future research.

C Proofs

Derivation of Equation (29) Recall, from equation (23), the expected utility from the optimal portfolio is

$$(\mathbf{m} - \boldsymbol{\pi})^\top \boldsymbol{\omega}^* - \tfrac{1}{2} A \boldsymbol{\omega}^{*\top} \boldsymbol{\Sigma} \boldsymbol{\omega}^*. \tag{50}$$

Equalizing the marginal benefit (the expected return) and the marginal cost (risk), the optimal portfolio in equation (27) is essentially the ratio of the expected return to risk.

Substituting the optimal portfolio in equation (27) into the first term above, we have

$$\begin{aligned}(\mathbf{m}-\boldsymbol{\pi})^\top \boldsymbol{\omega}^* &= (\mathbf{m}-\boldsymbol{\pi})^\top \sum_{i=1}^{K} \mathbf{u}_i \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{A\,\delta_i} \right) \\ &= \sum_{i=1}^{K} (\mathbf{m}^\top \mathbf{u}_i - \boldsymbol{\pi}^\top \mathbf{u}_i) \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{A\,\delta_i} \right) \\ &= \sum_{i=1}^{K} \frac{(\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi})^2}{A\,\delta_i} = \frac{1}{A} \sum_{i=1}^{K} \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{\sqrt{\delta_i}} \right)^2 .\end{aligned} \tag{51}$$

Notice, $\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}$ is a scalar and thus symmetric. Thus, the total expected return from the optimal portfolio is represented by the sum of squared Sharpe ratios of rotated portfolios, divided by risk aversion.

Now, we want to do the same thing to the second term in the expected utility.

$$\tfrac{1}{2} A \boldsymbol{\omega}^{*\top} \boldsymbol{\Sigma} \boldsymbol{\omega}^* \tag{52}$$

Here, since $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$, and $\boldsymbol{\Delta}$ is a diagonal matrix, we can express it as

$$\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top = \sum_{i=1}^{K} \delta\, \mathbf{u}_i \mathbf{u}_i{}^\top . \tag{53}$$

Also, $\mathbf{U}$ is an orthonormal matrix, which implies that $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$, an identity matrix. That is, $\mathbf{u}_i{}^\top \mathbf{u}_i = 1, \forall\, i$ and $\mathbf{u}_j{}^\top \mathbf{u}_i = 0, \forall\, j \neq i$. Then substituting the optimal portfolio, we have

$$\begin{aligned}\tfrac{1}{2} A \boldsymbol{\omega}^{*\top} \boldsymbol{\Sigma} \boldsymbol{\omega}^* &= \tfrac{1}{2} A \boldsymbol{\omega}^{*\top} \left(\sum_{i=1}^{K} \delta\, \mathbf{u}_i \mathbf{u}_i{}^\top \right) \left(\sum_{i=1}^{K} \mathbf{u}_i \left(\frac{\mathbf{u}_i{}^\top (\mathbf{m}-\boldsymbol{\pi})}{A\delta_i} \right) \right) \\ &= \tfrac{1}{2} A \boldsymbol{\omega}^{*\top} \sum_{i=1}^{K} \delta_i\, \mathbf{u}_i \left(\frac{\mathbf{u}_i{}^\top (\mathbf{m}-\boldsymbol{\pi})}{A\delta_i} \right) \\ &= \tfrac{1}{2} \boldsymbol{\omega}^{*\top} \sum_{i=1}^{K} \mathbf{u}_i \left(\mathbf{u}_i{}^\top (\mathbf{m}-\boldsymbol{\pi}) \right) \\ &= \tfrac{1}{2} \left(\sum_{i=1}^{K} \left(\frac{\mathbf{u}_i{}^\top (\mathbf{m}-\boldsymbol{\pi})}{A\delta_i} \right) \mathbf{u}_i{}^\top \right) \left(\sum_{i=1}^{K} \mathbf{u}_i \left(\mathbf{u}_i{}^\top (\mathbf{m}-\boldsymbol{\pi}) \right) \right) \\ &= \frac{1}{2A} \sum_{i=1}^{K} \left(\frac{\mathbf{u}_i{}^\top (\mathbf{m}-\boldsymbol{\pi})}{\sqrt{\delta_i}} \right)^2 .\end{aligned} \tag{54}$$

Thus, similar to the total expected return, the total risk from the optimal portfolio is

represented as the sum of squared Sharpe ratios of rotated portfolios, except that it is divided by 2 times the risk aversion. Thus, the total risk is exactly half of the total expected return, where half comes from the fact that the risk is a quadratic function of the portfolio, while the return is linear.

Finally, combining equations (51) and (54) yields equation (29).