ECONtribute
MARKETS & PUBLIC POLICY

ECONtribute
Discussion Paper No. 146

Flow Trading

Eric Budish
Jeongmin Lee
Peter Cramton
David Malec
Albert S. Kyle

February 2022

www.econtribute.de

UNIVERSITÄT BONN
Universität zu Köln

Funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – EXC 2126/1– 390838866 is gratefully acknowledged.

Flow Trading*

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

January 31, 2022
Preliminary Version

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 addresses six concerns with the current market design: (1) sniping and the speed race; (2) the complexities and inefficiencies caused by tick-size constraints; (3) the cost and complexity of trading large quantities over time, (4) of trading portfolios, and (5) of providing liquidity in correlated assets; (6) fairness and transparency of optimal execution.

*We thank Elizabeth Baldwin, Paul Klemperer, Paul Milgrom, David Parkes, and Marzena Rostek for helpful conversations. Disclosures: Peter Cramton consults on market design and is an academic advisor to Carta on the design of a private equity exchange. Albert S. Kyle has worked as a consultant for various U.S. government agencies on issues related to competition and efficiency in financial markets. He is a non-executive director of a U.S.-based asset management company. The other authors have no relevant or material financial interests that relate to this research. Research support: Budish thanks the University of Chicago Booth School of Business. Cramton thanks the German Research Foundation (DFG, EXC 2126/1-390838866) and the European Research Council 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 current 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. In some markets, 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—in a sense, tracing not only to the era of specialists and trading pits, but all the way back to trading under the buttonwood tree. A human can run a limit order book market with pen-and-paper or simple electronic recordkeeping if the orders arrive slowly enough, and computers can run limit order book 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.

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 price that just became stale. This 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.[2]

---

[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.

[2]See Budish, Cramton and Shim (2015) on the concept of sniping, how it differs from traditional adverse selection based on asymmetric private information (Kyle 1985, Glosten and Milgrom 1985), and its equilibrium effects on investments in speed and the market's cost of liquidity. See Aquilina, Budish and O'Neill (2022) for empirical evidence on the magnitude of sniping races.

Second, the discrete minimum tick size, which is necessary to prevent an explosion 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.[3]

Third, institutional investors trading large quantities of stock typically now do so by placing and canceling thousands of small orders, spread out over a period of time, to reduce price impact and disguise trading intents. 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.[4] 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 simply 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, they must 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 [30-50%] of all U.S. stock market volume.[5]

[3] 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.

[4] As one simple example, Hasbrouck and Saar (2013) pointed 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.

[5] Add source

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 until 1000 shares are bought" instead of "Buy a maximum of 1000 shares right now."

Second, instead of the market processing orders one at a time in sequence, orders are processed in discrete time using uniform-price batch auctions (i.e., “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 market-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 either 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. *That is, relative to the status quo market design, our proposed market design makes time discrete instead of continuous, and prices and quantities continuous instead of discrete.*

Third, instead of orders for a single asset, each order is for a portfolio of assets. A "portfolio" is a user-defined linear combination of assets, in which the asset weights can be arbitrary positive or negative real numbers. 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 opposite weights, the assets are substitutes because an increase in the price of one asset increases the quantity demanded of the other. For example, a pairs trade has a positive weight on the stock being bought and a negative weight on the stock being sold. An order to buy the S&P 500 has positive weights on each of the 500 stocks in the index. An order to sell a single asset, which represents an upward-sloping supply curve for the asset, is implemented as an equivalent downward sloping demand curve for a portfolio with a negative weight on the asset sold and zero weight on other assets.

Benefits Flow trading 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. Second, the various 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 incentive to race for advantageous queue position, further reducing the rents from speed. Third, investors who wish to trade large quantities over a period of time can do so directly, with a single order. They can easily tune the urgency of trade by choosing the maximum flow rate—e.g., 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 1985, Vayanos 1999, 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 in 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. Under flow trading, a market participant can directly provide liquidity in the pairs trades "Buy A, Sell B" and "Sell A, Buy B" (indeed, the latter is just an offer to sell the former). This means that even if an investor arrives wanting to buy just A, their trade can be automatically incorporated into the clearing prices of both A and B. There need not be a sniping race in asset B, nor is there any "correlation breakdown" of prices between A and B (Budish, Cramton, and Shim (2015)). The pairs trade order is like a string that ties the correlated assets' prices together, maintaining their underlying economic pricing relationships.

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 allows, for example, a retail investor who trades 100 shares over a minute, or an institutional investor who trades 100,000 shares over an hour, to infer the appropriate execution rate on their order from publicly announced market clearing prices exactly. An institutional investor trading a sophisticated portfolio can confirm directly that they received the correct execution. This perhaps should not sound radical, but it is a major transparency improvement over the current market design, where checking whether one's order received appropriate execution is very difficult (see Tyc (2014)).

Having mentioned these potential benefits, we add an important caveat, which is that flow trading is *not* designed to mitigate market failures related to market power or private information (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 micro-foundations for the bidding language.

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

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

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

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

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

1.1 Related Literature

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

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

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

Sophisticated expression of preferences over multiple objects is a theme in the market design literature more broadly. Research on this topic has straddled computer science, economics, and operations research (Lahaie and Parkes (2004); Sandholm and Boutilier (2006); Milgrom (2009); Klemperer (2010); Vohra (2011); Bichler (2017); Cramton (2017); Budish, Cachon, Kessler, and Othman (2017); Parkes and Seuken (2018);

Budish and Kessler (forthcoming)). This literature has mostly focused on indivisible-goods combinatorial allocation problems, such as spectrum auctions. Relative to this burgeoning literature, our contribution is our proposed language for portfolio orders, which treats all goods as perfectly divisible, and allows complementarities and substitutabilities only to the extent that they can be expressed with linear portfolio weights. This language is simple enough to obtain strong existence and computational results, while being expressive enough to capture many important use cases in financial markets.

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

A growing literature studies the implications of the status-quo market-design requirement that orders to trade an asset to be contingent only on the asset's own price and not on the price of other assets. In a competitive framework, Cespa (2004) studies price efficiency implications when traders instead can make their demands for a given asset contingent not only on the asset's own price but also on other asset prices. The more recent literature emphasizes the importance of strategic trading and price impact. Rostek and Yoon (2020*b*) and Wittwer (2021) find that such fully contingent demand can either increase or decrease welfare depending on market characteristics such as the size of the market and the correlation across assets. Rostek and Yoon (2020*c*) show that the welfare implications of introducing a new synthetic asset, like a portfolio of original assets, depend on price impact and symmetry across traders and assets. Chen and Duffie (2021) show that trading the same asset in multiple fragmented markets can improve welfare under some conditions. Antill and Duffie (forth.), on the other hand, find that fragmentation of financial markets generally harms welfare if some of the fragmented markets free-ride off of prices discovered on others of the fragmented markets (i.e., some markets engage in "size discovery" as distinct from "price discovery").

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

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

There are three main differences with the standard treatment, as explicated in MWG. First, the setting and assumptions are different.

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

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

3. While MWG require strongly monotone preferences and strictly positive prices, our preferences are not strongly monotone and prices can be negative. Individual assets can be "goods" or "bads". Moreover, it may be difficult to make preferences monotone, even over the restricted domain of agents' demands, because there is no natural "up" direction for the legs of a pairs trade.

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

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

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

A classic reference is Kelso Jr and Crawford (1982), who show that competitive equilibrium is guaranteed to exist in an indivisible goods setting under a substitutes condition. There have been many different variations of the Kelso-Crawford substitutes condition defined in the literature; see Gul and Stacchetti (1999); Milgrom (2000); Hatfield and Milgrom (2005); Ostrovsky (2008); Hatfield et al. (2013). Hatfield, Kominers, and Westkamp (2021) discusses the relationship among many of these criteria and provides a maximum domain result for existence.

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

Unlike in Baldwin and Klemperer (2019), or in most of the indivisible-goods substitutes literature, we obtain existence for *any* preferences expressible in our language.

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

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

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

2 Flow Orders

2.1 Formal Definition of Flow Orders

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

Flow orders depart from traditional limit orders in 3 ways:

1. Orders are for portfolios of assets instead of individual assets. A portfolio is defined by a vector of weights, $\mathbf{w}_i := (w_{i1}, \ldots, w_{iN})^\top$, where i identifies the order, N denotes the number of assets in the market, and $w_{in} \in \mathbb{R}$ denotes the portfolio weight of asset n in order i. A strictly positive weight denotes buying the asset, a strictly negative weight denotes selling the asset, and a zero weight denotes that that the asset is not a part of that portfolio.

2. Instead of step-wise demand, flow orders specify piecewise-linear downward-sloping demands. The user 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, and interprets p_i^H as indicating zero demand for the portfolio at prices weakly higher than p_i^H. Then, in the interval $[p_i^L, p_i^H]$, the flow order linearly interpolates the quantity demanded from full quantity at p_i^L to zero quantity at p_i^H.[6] Note that we use the phrase "buy the portfolio" to include the case of selling assets—in our language, selling an asset is buying a portfolio with a negative weight on the asset at a negative price (i.e., receiving a transfer). We will clarify this point, which can be confusing at first, 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

[6] 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.

quantities express a demand to trade immediately (1000 shares right now!) the user can tune their urgency to trade.

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

Next we formally 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 $Q_i^{\max}$.[7] 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 both the order's quantity limit q_i and where the price for the portfolio is relative to the order's price parameters p_i^L and p_i^H. If the portfolio price p_i is less than or equal to p_i^L, the order is "fully executable" and the portfolio is bought at the maximum rate q_i. If the portfolio price p_i is higher than p_i^H, then the order is "nonexecutable" and does not buy at all. If the portfolio price is somewhere between p_i^H and p_i^L, then the order is "partially executable" and buys at the rate determined by linear interpolation between the two price parameters.

Buying vs. Selling This formulation treats "selling" an asset as buying a portfolio with a negative weight on that asset at a negative price. This not only generates compact notation for representing both buying and selling but also emphasizes a symmetry between buying and selling which will be important for understanding how market clear-

[7] 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 $Q_i^{\max} - Q_i^t$.

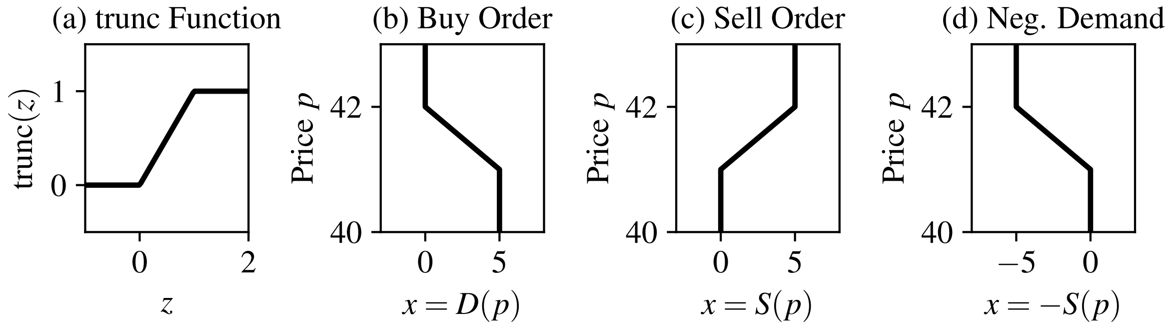

Figure 1: Plots of (a) the function $\operatorname{trunc}(z)$; (b) a single buy order, with pricing parameters p_i^L = \$41.00 and p_i^H = \$42.00, and maximum flow demand of q_i = 5.00 portfolio units per batch auction; (c) a single sell order, initially plotted as an upward-sloping supply curve with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

ing works. General equilibrium theory often uses this idea that an upward sloping supply curve for positive quantities is equivalent to a downward sloping demand curve for negative quantities.

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

See Figure 1 for an illustration of buying and selling.

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

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

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

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

2.2 Key Examples

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

1. Standard limit order.

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

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

   In the traditional market design, a market order executes immediately at the clearing price. The analog here is a time-weighted average price (TWAP) order. The user specifies a price parameter p_i^L that is sufficiently aggressive relative to recent prices that it is esssentially guaranteed to execute.[8] Then, the user will trade quantity q_i of the portfolio every batch auction until their quantity limit is achieved, i.e., they will trade at the TWAP over this time period.

3. Pairs trades.

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

4. Portfolio trades.

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

   We note that traders can construct and trade their own index portfolios. For example, an order to buy the S&P 500 has positive weights on each stock in the S&P 500 index, with weights proportional to S&P 500 weights and zero weight on stocks not in the S&P 500 index. An order to sell an index has negative weights on all stocks

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

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 ESG constraints.

5. General long-short strategies.

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

6. Market making strategies.

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

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

2.3 Limitations of the Language

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

First, trading demands are only defined at exactly the ratio of portfolio weights specified in the order. If an order specifies it wants to buy assets A and B at a ratio of 2:1, the order contains no information about the trader's willingness to trade at, say, a ratio of 2.2:1 or 1.8:1. This restriction relative to traditional consumer theory, where preferences are typically defined on the whole positive orthant, is key to our method of existence proof (below in Section 3.2).

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 do take advantage of this linearity. We view the linearity restriction as less important a limitation than some of the others, because arbitrary downward-sloping functions can be approximated, if needed, with a set of linear orders.

Third, the language does not allow for indivisibilities. Most importantly, a user cannot specify a minimum transaction quantity per batch, only a maximum. So, for example, an order cannot be "fill or kill", or "at least 100 shares per batch, otherwise stay out".

That said, a user may be able to approximate such preferences with marketable orders if prices are continuous enough.

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

3 Market Clearing Prices and Quantities

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

3.1 Definition of Market Clearing

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

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

The function $D(\cdot)$ 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 $D_i(\boldsymbol{\pi}^\top \mathbf{w}_i)\mathbf{w}_i$, 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 $\mathbf{q} = D(\boldsymbol{\pi})$ may have both positive and negative components. Note as well 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.

3.2 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.

In the range of prices where an order is partially executable, the demand is a linear function of prices. Therefore, a quadratic quasilinear utility function defines preferences. The constraints preventing overfilling or underfilling the order are linear inequality constraints. The constraint that markets clear are linear equality constraints. Putting this together mathematically results in the problem of maximizing a quadratic 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 prove that clearing prices and quantities exist.

Pseudo-Utility Let $V_i(x)$ denote the dollar utility of order i from a trade rate of x in portfolio units per second. To find $V_i(x)$, we first define the marginal utility function $M_i(x)$ as the inverse demand curve, $p_i = M_i(x_i)$, where recall the order i demand curve is denoted by $D_i(p_i) = x_i$. In words, the inverse demand curve maps order i's trade rate

$x \in [0, q_i]$ into prices $p \in [p_i^L, p_i^H]$.[9] Rearranging equation (2) we have:

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

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

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

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

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

We will think of $V_i(x)$ as defined for all $x \in \mathbb{R}$, with order specifications imposing the constraint $x \in [0, q_i]$.[10]

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

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

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

subject to choosing quantities consistent with market clearing constraints and order ex-

[9] 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.

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

ecution rate constraints:

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

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

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

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

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

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

Proof. The problem has the following properties:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4. Feasibility: No trade ($\mathbf{x} = \mathbf{0}$) is feasible because it clears the markets and is allowed on each order.[11]

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

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

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.

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

4 Computation

In this section we study computational feasibility of flow trading. The objective here 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 and a publicly available algorithm. For this, we simulate order books and find that solutions can be found in less than a quarter of a second for a wide range of parameters. Section 4.1 proposes a computational methodology, and Section 4.2 explores computational performance in a simulation environment.

4.1 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 a direction proportional to net excess demand, and the process continues until the market clears. Tatonnement is equivalent to using the gradient method of optimization when there is a function of prices whose first order conditions correspond to market clearing. In our setting, such a function can be found by minimizing out the multipliers μ and λ in the dual objective function in equation (12), which we call the gains function. Theorem 2 implies that the function's first order condition corresponds to market clearing.

Since the gains function has a piecewise-linear derivative, it is continuously differentiable, and the derivative satisfies a Lipschitz condition[12]. These conditions assure that the gradient method converges (Nesterov (2004, Corollary 2.1.2, p. 70)). However, while the guaranteed convergence rate is much faster than for the traditional general-equilibrium theory problems discussed by Scarf and Hansen (1973)[13], 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 very frequently throughout the trading day (as opposed, e.g., to a single high-stakes allocation problem like in combinatorial auctions).

[12] $|\nabla G(\boldsymbol{\pi} + \Delta\boldsymbol{\pi}) - \nabla G(\boldsymbol{\pi})| < L|\Delta\boldsymbol{\pi}|$ for some Lipschitz constant L.

[13] 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.

4.1.1 Interior Point Method

We propose to use an interior point method for quadratic programming. The literature shows that interior point methods are computationally more efficient than the intuitive gradient method, both theoretically (see Nesterov (2004, Chapter 4); Bertsekas (2009), Boyd and Vandenberghe (2004)) and in practice (Gondzio (2012)).[14]

Exchange as a Small Market Maker Theoretically, the interior point method requires the existence of an interior point, 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, however, there is no natural candidate for such an interior point. For example, no-trade ($\mathbf{x} = \mathbf{0}$), which satisfies market clearing, does not lie on the interior of the constraints.

To ensure an interior point, we introduce the possibility that the exchange acts as a very small market maker for every asset. This makes allocations feasible by taking into its inventory otherwise uncleared quantities. Specifically, the exchange submits a linear demand curve for each asset n:

$$\epsilon_n(\pi_{0n} - \pi_n), \tag{15}$$

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 very small positive number such that the exchange trades little. 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 the exchange as a small market maker, existence of an interior point is easily assured. For example, pick any $\mathbf{x}$ such that $\mathbf{0} < \mathbf{x} < \mathbf{q}$. Then the exchange can soak up any uncleared quantity to clear the market.

Further, allowing modest exchange trading has two other benefits. First, it can resolve the "tiebreaker problem", in case there is an interval of market-clearing prices (as we know is possible from Theorem 2). Since the exchange has an active order for every asset at every relevant price vector, market prices are chosen uniquely for all assets when multiple prices are possible otherwise. For example, if π_{0n} is set at the previous clearing

[14]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 problems.

price for asset n, then the exchange's small trading demand will break ties in favor of the price closest to the previous price. Second, it can absorb uncleared quantities due to rounding error and inexact convergence of the algorithm for calculating clearing prices, even if the algorithm has practically "converged" to a target tolerance.

Solving the KKT Conditions To find market clearing prices and quantities, the interior point method solves the Karush–Kuhn–Tucker (KKT) conditions, 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 all of the results from Section 3 still 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 every price, the solution is unique.

Theorem 3 (Karush–Kuhn–Tucker (KKT) Conditions with Exchange Trading)**.** *Any solution of the KKT conditions* (16)–(19) *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:*

$$\mathbf{W}\mathbf{x}^* = \mathbf{0}, \qquad \mathbf{0} \le \mathbf{x}^* \le \mathbf{q}, \qquad \textit{(Primal Feasibility)} \tag{16}$$

$$\boldsymbol{\pi}^* \in \mathbb{R}^N, \qquad \boldsymbol{\lambda}^* \ge \mathbf{0}, \qquad \boldsymbol{\mu}^* \ge \mathbf{0}, \qquad \textit{(Dual Feasibility)} \tag{17}$$

$$\mathbf{p}^H - \mathbf{D}\mathbf{x}^* - \mathbf{W}^\top \boldsymbol{\pi}^* + \boldsymbol{\mu}^* - \boldsymbol{\lambda}^* = \mathbf{0}, \qquad \textit{(Primal Optimality)} \tag{18}$$

$$\boldsymbol{\lambda}^* \cdot (\mathbf{q} - \mathbf{x}^*) = \mathbf{0}, \qquad \boldsymbol{\mu}^* \cdot \mathbf{x}^* = \mathbf{0}, \qquad \textit{(Complementary Slackness)} \tag{19}$$

With exchange trading defined in equation (15)*, 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; see Bertsekas (2009, Theorem 5.34(b), p. 173). Uniqueness follows from the exchange having a partially executable order for every asset. 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 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 (19) with a set of constraints parameterized by $\bar{\nu} > 0$:

$$\boldsymbol{\lambda}^* \cdot (\mathbf{q} - \mathbf{x}^*) = \bar{\nu} \cdot \mathbf{1}, \qquad \boldsymbol{\mu}^* \cdot \mathbf{x}^* = \bar{\nu} \cdot \mathbf{1}. \tag{20}$$

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

The modified complementary slackness conditions in equation (20) 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 The algorithmic strategy is to solve the modified KKT conditions (16), (17), (18), and (20) iteratively by starting with an initial guess for $\mathbf{x}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$. On each iteration, we substitute $\mathbf{x} + \Delta\mathbf{x}$ for $\mathbf{x}$, $\boldsymbol{\pi} + \Delta\boldsymbol{\pi}$ for $\boldsymbol{\pi}$, $\boldsymbol{\mu} + \Delta\boldsymbol{\mu}$ for $\boldsymbol{\mu}$, and $\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$ for $\boldsymbol{\lambda}$, then solve the linearized system for $\Delta\mathbf{x}$, $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\mu}$, and $\Delta\boldsymbol{\lambda}$ with the value of $\bar{\nu}$ set to 0. To keep the new guess an interior point, the solution vectors are multiplied by a scalar α (with $0 < \alpha \leq 1$) to insure 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. 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.[15]

[15]The revised KKT system is nonlinear in the unknowns $\boldsymbol{\pi}$, $\mathbf{x}$, $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$, and $\bar{\nu}$ only because the revised complementary slackness condition involves element-by-element multiplication of $\mathbf{x}$ by $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$. For $\boldsymbol{\mu}$ (and analogously for $\boldsymbol{\lambda}$), linearizing $(\mathbf{x} + \Delta\mathbf{x})(\boldsymbol{\mu} + \Delta\boldsymbol{\mu})$ sets the second-order term term $\Delta\mathbf{x}\Delta\boldsymbol{\mu}$ to zero. To correct for the error created by dropping the second-order term, 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.

Our solution uses our own straightforward Python implementation of the interior point methodology in the CVXOPT package, as described by Vandenberghe (2010). Both the Python programming language and the CVXOPT package are free and publicly available. One version of the algorithm is implemented on the cpu using the Python packages numpy and scipy. Another equivalent 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 is tailored to our specific

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 asset	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 ord 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.

The positive definite matrix to be decomposed changes with each iteration because it is constructed by implicitly assigning weights to each order based on values of multipliers. The weights are close to zero 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 far away from the boundary of the interval $[0, q_i]$. The order is expected to be relevant for price discovery at the margin when it is partially executable. A new Cholesky decomposition is needed on each iteration to incorporate updated weights from the most recent iteration into calculation of the new search direction.

4.2 Results

4.2.1 Simulating the Order Book

There are four sets of parametric assumptions used to simulate an "order book." A list of parameters is presented in Table 1, with "base case," high, and low values.

The first set of assumptions includes the number of assets, the number of orders, and the extent of exchange trading. As a base case, we start with 500 assets and 100,000 orders. The number 500 is chosen based on the number of stocks in the S&P 500 index.

quadratic program, which has an invertible diagonal matrix **D** and simple "Euclidean cone" constraints $\mathbf{0} \leq \mathbf{x} \leq \mathbf{q}$. Our implementation code will be posted publicly upon publication, and is available immediately to interested readers upon request.

The number 100,000 is chosen arbitrarily. After reporting outcomes for the base case, we study how the number of assets and the number of orders affect computation.

Exchange trading is characterized by the slope of its demand schedule (ϵ_n in Equation (15)) and the base price. For simplicity, we set the slope constant across all assets, and choose a small number, 10^{-2}, so that the exchange does very little trading, buying one dollars worth of an asset when the price falls by one percent. The base price, at which the exchange starts buying and selling all assets, is set equal to the initial price. The initial price of all assets and index portfolios is normalized to \$100. These prices are pure scaling factors, which do not affect the economics of market clearing in any way.

The second set of assumptions defines how orders are distributed across individual assets and various types of portfolios and how orders are distributed between buy and sell. Theoretically, there are infinitely many different portfolios which investors can choose from by combining any of the 500 assets with arbitrary weights for each asset. In our simulations, however, we restrict portfolios to six different types of index portfolios and to randomly generated pairs trades. This limitation on portfolios lowers the computational cost of our simulations.

For index portfolios, we construct value-weighted and equal-weighted portfolios of the market index, "size" indices, and "industry" indices. Assets are sorted by expected dollar volumes to be assigned to size quantiles, and the groups created form size indices. Assets are evenly allocated to industries by first sorting assets by size, then assigning assets to the same industry if they have the same rank modulo the number of industries. Pairs trades randomly buy either an asset or an index portfolio and sell an equal expected dollar value of another asset or index.

In the base case scenario, we divide the 100,000 orders evenly between orders for individual assets and orders for portfolios. The orders for portfolios are then divided evenly between index portfolios and pairs trades. Orders for index portfolios are randomly assigned to the 6 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, except to reflect that the value-weighted market indices, such as those which track S&P 500 (including even the CME's S&P 500 E-mini contract), have high trading volume. We later vary the probabilities to study how they may affect computation times.[16] Finally, each order for individual assets and indexes has an equal probability of being buy or sell.

The third set of assumptions is on the distribution of the number of orders across assets and the size of orders. To represent the skewness of the trading volume in the real-world stock market, we allow the number of orders and the size of orders to be drawn from heterogeneous distributions.

For each asset, we draw a random number from a lognormal distribution with mean of 1 and 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.

The size of orders for a given asset is lognormally distributed with standard deviation of 1.5. Following the market microstructure invariance hypothesis of Kyle and Obizhaeva (2016), the mean is proportional to the square root of the expected number of orders for that asset. Specifically, the mean equals $k * \sqrt{\text{expected number of orders}}$, where k is a constant chosen to make the aggregate expected order volume from individual stocks equal to the arbitrary scaling constant of \$10 million per second using arbitrary expected ex ante prices of \$100 per share.

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 orders by the same factor k 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 largely 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.

The last set of assumptions is on limit prices. First, the upper limit price (p_i^H) is log-

[16] To allow varying the these probabilities in a convenient manner, 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 a equal-weighted size (industry) index portfolio. The five parameters, together with the restriction that the probabilities should sum to one, determine all six probabilities. We let each of the five parameters to vary from 5% to 95%.

normally distributed with the mean of $97 if buying and -$103 if selling, and the standard deviation of 10% of the initial price. Second, the difference between the upper and lower limit prices, $p_i^H - p_i^L$, is lognormally distributed with mean of 1 basis point (relative to the "midpoint" $(p_i^L + p_i^H)/2$ and a standard deviation of 2. The very small mean of one basis point and the large variance of 2 are expected to stress the algorithm by making demand highly nonlinear in prices. All of the random variables are independently distributed.

4.2.2 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.[17] Uncleared quantities are basically zero, equal to a fraction 8.7e-12 of total volume, or or 8.7 dollars per trillion dollars.

The amount of exchange trading is small. On average, it trade 3.2 dollars per million dollars of trading volume. Across 51 repetitions, the maximum, the minimum, and the standard deviation of exchange trading are 5.99, 2.32, and 0.73 dollars per million dollars, respectively. 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 large effect on the cross-sectional standard deviation of market-clearing prices for assets with very thin realized order books. Without exchange trading, arbitrary determination among multiple prices drives the standard deviation to a ridiculously large value of 1.44e+07%. Small exchange trading brings it down to a much more reasonable 14.19%. In practice, we would expect market-making firms to provide liquidity in thinly traded stocks and stabilize prices, possibly at a wide spread.

Overall, market clearing allocations are computed quickly and accurately and prices are reasonably stable with minimal trading by the exchange. We interpret these results as being a sufficient proof of concept for the flow trading market design when the market

[17] Computation times do not change much when more cores are used. This is probably because easily parallelized computations are done on the gpu while not easily parallelized computations, like Cholesky decomposition and some sparse matrix operations, are done on the cpu but do not benefit much from using multiple cores. The computation times are stable across 401 repetitions, with the maximum of 0.1603 seconds, the minimum of 0.1365 seconds, and a standard deviation of 0.0058 seconds.

clears at intervals of once per second.

Next, we study how computation times vary with the number of assets and the number of orders; see Figure 2. In the first panel, as the number of orders increases from 100,000 to 1,000,000 and 3,000,000, while keeping the number of assets constant, computation times increase from 0.1451 to 0.5639 and 1.5207 seconds, respectively. 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, respectively. The computation time crosses one second and ten seconds with approximately 1,800 assets and 5,200 assets, respectively. The increased computation times when the number of assets increase are mainly due the computation costs of calculating the matrix decomposed by the Cholesky decomposition and the Cholesky decomposition itself (which is an $O(N^3)$ algorithm in the number of assets). 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.

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 modest effect on computation times except for the two parameters: 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 buyers from 0.5 to 0.1 increases to 0.2344 seconds. The large values for these parameters make the order book more asymmetric, and it becomes more difficult to solve for market clearing.

Finally, we consider an extreme scenario by settings 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

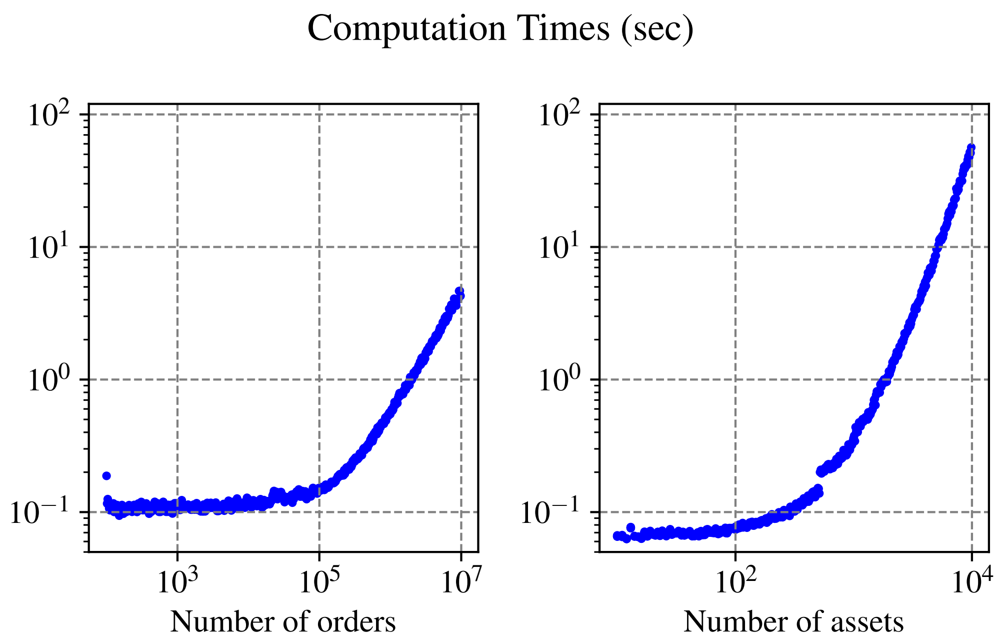

Figure 2: Computation Times Panel A varies the number of orders, holding the number of assets fixed at the baseline value of 500. Panel B varies the number of assets, holding the number of orders fixed at the baseline value of 100,000. In both panels, all other parameters are set to their baseline values. Each dot represents one simulation, and there are 500 simulations in each panel. The small discontinuity in Panel B around 600 assets is caused by [we don't know yet.]

half a second. The analysis suggests computation times are not sensitive to the specific parameter values used for the order book construction.

Discussion In a production environment in the future, we expect more powerful computers and more refined algorithms will make it easier to calculate market clearing allocations with even greater speed. More cores, faster CPU, and more memory bandwidth will all likely reduce the computation times. Quadratic programming and sparse matrix multiplications, which play key roles in our computation, are an active area of research in computer science. Developments in these areas will also lead to more efficient computation. Further, given that our results are from using a single core of the workstation, refining the algorithm to facilitate parallel processing may be able to dramatically reduce computation times.

5 Microfoundation for Orders for Portfolios

Flow orders as defined in this paper specify demand for a user-specified portfolio as a function of the price of that portfolio. While such orders for portfolios are more general than traditional limit orders, this language is still restrictive, as in general a market participant's asset demands depend on the complete vector of asset prices, not just on the price of a user-specified portfolio. In this section, we provide a modest microfoundation for this paper's approach to expressing trading demands.

5.1 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 with orders for portfolios. The model is static so there is no distinction between trading in quantities and trading in flows; we will discuss this model interpretation issue in greater detail below. 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)). These models focus on the case of a single risky asset, but 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 $\boldsymbol{\Sigma}$. The trader has CARA preferences with risk aversion parameter A. There are no wealth effects with CARA preferences, so for simplicity set the trader's wealth to zero.

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 as well 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:

$$\max_{\boldsymbol{\omega}} \mathrm{E}\left[-\exp^{-A(\mathbf{v}-\boldsymbol{\pi})^\top \boldsymbol{\omega}}\right], \tag{21}$$

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

$$\max_{\boldsymbol{\omega}} \left[(\mathbf{m} - \boldsymbol{\pi})^\top \boldsymbol{\omega} - \frac{1}{2} A\, \boldsymbol{\omega}^\top \boldsymbol{\Sigma} \boldsymbol{\omega} \right]. \tag{22}$$

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

$$\boldsymbol{\omega}^* = (A\, \boldsymbol{\Sigma})^{-1} (\mathbf{m} - \boldsymbol{\pi}). \tag{23}$$

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 in general 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 (23) with a single portfolio order with portfolio weights $\mathbf{w}_i$ and quantity parameter $Q_i^{\max}$ such that $\mathbf{w}_i Q_i^{\max} = \boldsymbol{\omega}^*$. 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 case might capture, for example, that prices are changing over time and traders trade gradually. We next show that traders can implement their optimum according to (23) with portfolio orders, without any knowledge of prices. To do this, we need to “rotate” the asset space such that it is spanned by independent portfolios.

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

$$\boldsymbol{\Sigma} = \mathbf{U} \boldsymbol{\Delta} \mathbf{U}^\top, \tag{24}$$

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 $\mathbf{U}$.[18] Then we have

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{K} \frac{1}{\delta_i} \mathbf{u}_i \mathbf{u}_i^\top. \tag{25}$$

[18]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 (23) as

$$\boldsymbol{\omega}^* = \sum_{i=1}^{K} \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{A\,\delta_i} \right) \mathbf{u}_i, \tag{26}$$

which is a combination of demand schedules for portfolios. Here, $\mathbf{u}_1, \ldots, \mathbf{u}_K$ are portfolios of assets, whereas in (23) demand was expressed in terms of individual assets. Since the portfolios are independent of one another (and there is no wealth effect in CARA preferences), the optimal portfolio chooses the demand for each of them separately as if in a single-asset model.[19] That is, the optimal demand for the ith portfolio is given by

$$\frac{1}{A\,\delta_i} (\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}), \tag{27}$$

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 $\mathbf{u}_i$, respectively. Since the demand for each portfolio only depends on the portfolio's price, traders can achieve the optimal trade in equation (23) by utilizing K orders for portfolios where each order is a function of that portfolio's price.

Recall, in our proposed market design, we require orders' demands for portfolios to be downward sloping. Since the optimal demand for each portfolio in equation (26) 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* (23)*) 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* (26)*).*

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 (26) into equation (22), and some algebraic manipulations (see details in

[19] Observe that in equation (23), 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

$$\sum_{i=1}^{K} \frac{1}{2A} \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{\sqrt{\delta_i}} \right)^2. \tag{28}$$

This shows that the benefit of each portfolio is determined by its squared Sharpe ratio as perceived by the trader.[20] In practice, traders may select a few portfolios, which they perceive to have a sufficiently high Sharpe ratio (more precisely, its absolute value), and choose to trade only those portfolios rather than all of the K portfolios.

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 of course move prices, and 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, taking into account their price impact.

Following the literature (for example, Kyle (1989); Malamud and Rostek (2017)), we assume that traders believe that their price impact is linear in the quantity they trade. We further assume that the matrix of price impact is positive semidefinite.[21] That is, for each trader, there is an $N \times N$ positive semidefinite matrix $\boldsymbol{\Lambda}$, such that

$$\boldsymbol{\pi} = \boldsymbol{\pi}_0 + \boldsymbol{\Lambda}\, \boldsymbol{\omega}, \tag{29}$$

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 com-

[20]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.

[21]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.

petitive solution in equation (23), given by

$$\boldsymbol{\omega}^* = (A\,\boldsymbol{\Sigma} + \boldsymbol{\Lambda})^{-1}(\mathbf{m} - \boldsymbol{\pi}). \tag{30}$$

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 it is spanned by independent portfolios. Then the same logic as above implies that the optimal portfolio can be implemented by combining portfolio orders that only depend on the portfolio's price. The number of required portfolio orders corresponds to the rank of $A\,\boldsymbol{\Sigma} + \boldsymbol{\Lambda}$.

Theorem 5. *Consider a static CARA-normal framework in which a trader believes that her price impact is linear and positive semidefinite (equation* (29)*). Then the strategic trader's optimal portfolio (equation* (30)*) can be represented as the sum of downward-sloping demand schedules for portfolios, each of which depends only on that portfolio's price.*

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

5.2 Approximations for General Preferences and Limitations

Our logic in the previous subsection extends to an arbitrary strictly concave twice continuously differentiable quasilinear utility function over assets, provided that asset payoffs are joint normally distributed. To see why this is the case, recall, the two key properties of CARA preferences we use in the arguments above are one, that there are no wealth effects, and two, that they are strictly concave.

First, generally 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, for any quasilinear preferences, which imply no wealth effect, we can use the independent portfolios such that the trader's optimal demand for each portfolio only depends on that portfolio's price.

Second, the strict concavity implies that the optimal demand for any portfolio must be downward sloping. This is crucial since we require demands for portfolios to always be downward sloping in the portfolio's price. While the optimal strategy may not be linear for an arbitrary strictly concave twice continuously differentiable quasilinear preference, we can approximate the optimal strategy by combining multiple linear downward sloping schedules.

However, portfolio orders will not be able to approximate the optimal portfolio of every concave utility function closely. 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 with each other. In this case, the optimal demand for an independent portfolio may again depend on the prices of other portfolios.

6 Discussion of Implementation and Policy Issues

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

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

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

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

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

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

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

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

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

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

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

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

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

7 Conclusion

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

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

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

An open topic left for future research is the efficiency and welfare consequences of portfolio trading. We conjecture there are two main efficiency benefits. First, complexity and cost benefits of allowing market participants to directly express many common trading demands, which reduces systems complexity and the need for costly intermediation. Second, flow 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.

References

Admati, Anat R. 1985. "A noisy rational expectations equilibrium for multi-asset securities markets." *Econometrica: Journal of the Econometric Society*, 629–657.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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." *The 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.

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." *The 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: Journal of the Econometric Society*, 1483–1504.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

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

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

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

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

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

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

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

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

Appendix

A Proofs

Derivation of Equation (28) Recall, from equation (22), the expected utility from the optimal portfolio is

$$(\mathbf{m}-\boldsymbol{\pi})^\top \boldsymbol{\omega}^* - \frac{1}{2} A\, \boldsymbol{\omega}^{*\top} \boldsymbol{\Sigma} \boldsymbol{\omega}^*. \tag{31}$$

Equalizing the marginal benefit (the expected return) and the marginal cost (risk), the optimal portfolio in equation (26) is essentially the ratio of the expected return to risk.

Substituting the optimal portfolio in equation (26) into the first term above, we have

$$\begin{aligned}
(\mathbf{m}-\boldsymbol{\pi})^\top \boldsymbol{\omega}^* &= (\mathbf{m}-\boldsymbol{\pi})^\top \sum_{i=1}^{K} \mathbf{u}_i \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{A\,\delta_i} \right) \\
&= \sum_{i=1}^{K} (\mathbf{m}^\top \mathbf{u}_i - \boldsymbol{\pi}^\top \mathbf{u}_i) \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{A\,\delta_i} \right) \\
&= \sum_{i=1}^{K} \frac{(\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi})^2}{A\,\delta_i} = \frac{1}{A} \sum_{i=1}^{K} \left(\frac{\mathbf{u}_i{}^\top \mathbf{m} - \mathbf{u}_i{}^\top \boldsymbol{\pi}}{\sqrt{\delta_i}} \right)^2 .
\end{aligned} \tag{32}$$

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.

$$\frac{1}{2} A \boldsymbol{\omega}^{*\top} \boldsymbol{\Sigma} \boldsymbol{\omega}^* \tag{33}$$

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

$$\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top = \sum_{i=1}^{K} \delta \mathbf{u}_i \mathbf{u}_i{}^\top. \tag{34}$$

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

$$
\begin{aligned}
\frac{1}{2}A\boldsymbol{\omega}^{*\top}\boldsymbol{\Sigma}\boldsymbol{\omega}^{*} &= \frac{1}{2}A\boldsymbol{\omega}^{*\top}\left(\sum_{i=1}^{K}\delta\mathbf{u}_i\mathbf{u}_i^{\top}\right)\left(\sum_{i=1}^{K}\mathbf{u}_i\left(\frac{\mathbf{u}_i^{\top}(\mathbf{m}-\boldsymbol{\pi})}{A\delta_i}\right)\right) \\
&= \frac{1}{2}A\boldsymbol{\omega}^{*\top}\sum_{i=1}^{K}\delta_i\mathbf{u}_i\left(\frac{\mathbf{u}_i^{\top}(\mathbf{m}-\boldsymbol{\pi})}{A\delta_i}\right) \\
&= \frac{1}{2}\boldsymbol{\omega}^{*\top}\sum_{i=1}^{K}\mathbf{u}_i\left(\mathbf{u}_i^{\top}(\mathbf{m}-\boldsymbol{\pi})\right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{K}\left(\frac{\mathbf{u}_i^{\top}(\mathbf{m}-\boldsymbol{\pi})}{A\delta_i}\right)\mathbf{u}_i^{\top}\right)\left(\sum_{i=1}^{K}\mathbf{u}_i\left(\mathbf{u}_i^{\top}(\mathbf{m}-\boldsymbol{\pi})\right)\right) \\
&= \frac{1}{2A}\sum_{i=1}^{K}\left(\frac{\mathbf{u}_i^{\top}(\mathbf{m}-\boldsymbol{\pi})}{\sqrt{\delta_i}}\right)^2.
\end{aligned} \tag{35}
$$

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 (32) and (35) yields equation (28).

B Pete's Tables

Table 3 shows the results of Mina's exchange trading model for the base case model with 500 assets and 30000 orders. Each row represents averages across the same 101 orders books simulated on PK's laptop (except that exchange orders obviously differ for each row in each simulation). The parameter $epsilon$ has the same meaning as ϵ in the paper: number of shares bought or sold by the exchange per one dollar change in price. This number is calculated by multiplying the notebook variable $frac\_exch\_liquidity$ by 10^{-4}. The variable $maxvfrac$ defines q for the exchange as the maximum fraction of expected volume bought or sold. The variables $unclrdpM$ and $exchpM$ define the amounts of uncleared volume as exchange volume as dollars per one million dollars of total market volume (defined by multiplying the basis-point numbers in the notebook by 100). The variable $maxnumq$ is the average of the number of assets for which the exchange trades the maximum quantity allowed.

Here is PK's interpretation of the results:

Uncleared volume, which perhaps captures numerical error and convergence tolerances, does not vary much across rows.

For very small values $\epsilon = 10^{-16}, 10^{-14}, 10^{-12}$, the standard deviation of prices and the number of iterations both increase if the exchange trades more, but uncleared volume does not decrease much. There are evidently some poorly conditioned relationships among asset prices, perhaps due to illiquid assets being demanded by indexes but not supplied with much natural liquidity in the individual asset itself. Perhaps more exchange trading helps define these asset prices in a manner which destabilized the prices and takes more time to compute. Clearly, for the smallest values of exchange trading, the exchange is playing the role of tiebreaker.

For small values $epsilon = 10^{-16}, 10^{-14}, 10^{-12}, 10^{-10}$, the number of iterations decreases when the exchange trades more, but the exchange does not trade much more than rounding error when $maxvfrac$ is small. This validates our claim that a small amount of exchange trading improves numerical efficiency without requiring much inventory accumulation. A good choice of parameters might be $epsilon = 10^{-10}$, $maxqfrac = 0.10$ or 1.00 (rows 19, 20), where convergence occurs in about 30 or 31 iterations and exchange trading is about one order of magnitude greater than rounding error.

If $epsilon > 10^{10}$, the number of iterations can be reduced slightly to 28 or 29, but this larger value of epsilon requires more exchange trading.

When the exchange trades a large amount ($epsilon > 100$), exchange trading noticabley dampens price changes, and the exchange frequently reaches its maximum trading limit when $maxvfrac$ is small. As a sanity check, when $epsilon$ and $maxvfrac$ are both very large, the exchange—as expected—stabilizes prices at 100 percent of their target with a standard deviation close to zero. If this did not happen, it would indicate a bug in PK's logic.

Table 3: *** Mina's exchange model ***

	epsilon	maxvfrac	dt	stddt	niter	unclrdpM	exchpM	pmeanpct	pstdpct	maxnumq
0	1.00e-16	1.00e-02	0.9405	0.1225	36	1.10e-02	4.27e-04	99.39	1.37e+05	0.00
1	1.00e-16	1.00e-01	0.9694	0.1885	38	1.21e-02	4.26e-03	99.39	2.06e+05	0.00
2	1.00e-16	1.00e+00	1.0691	0.2280	42	1.21e-02	4.26e-02	99.39	4.44e+05	0.00
3	1.00e-16	1.00e+02	1.5621	0.4065	63	1.53e-03	4.26e+00	99.39	1.20e+06	0.00
4	1.00e-16	1.00e+03	2.1272	0.4708	85	1.86e-03	4.26e+01	99.39	3.20e+06	0.00
5	1.00e-16	1.00e+04	2.4090	0.4540	99	4.75e-03	4.26e+02	99.39	6.91e+06	0.00
6	1.00e-14	1.00e-02	0.9424	0.1766	37	6.73e-03	4.27e-04	99.39	1.14e+04	0.00
7	1.00e-14	1.00e-01	0.9285	0.1781	36	8.30e-03	4.26e-03	99.39	1.15e+04	0.00
8	1.00e-14	1.00e+00	0.9544	0.2013	38	4.97e-03	4.26e-02	99.39	1.15e+04	0.00
9	1.00e-14	1.00e+02	1.2281	0.4264	47	3.46e-04	4.26e+00	99.39	1.43e+04	0.00
10	1.00e-14	1.00e+03	1.4666	0.4801	59	1.02e-04	4.26e+01	99.39	1.77e+04	0.00
11	1.00e-14	1.00e+04	1.7127	0.4767	69	6.20e-05	4.26e+02	99.39	2.29e+04	0.00
12	1.00e-12	1.00e-02	0.8188	0.0588	32	3.63e-03	6.15e-04	99.39	9.79e+03	0.00
13	1.00e-12	1.00e-01	0.8134	0.0636	32	2.71e-03	4.47e-03	99.39	9.79e+03	0.00
14	1.00e-12	1.00e+00	0.8236	0.0434	32	2.83e-03	4.27e-02	99.39	9.79e+03	0.00
15	1.00e-12	1.00e+02	0.9306	0.0647	36	7.18e-04	4.26e+00	99.39	9.79e+03	0.00
16	1.00e-12	1.00e+03	0.9916	0.0623	39	4.42e-04	4.26e+01	99.39	9.79e+03	0.00
17	1.00e-12	1.00e+04	1.0914	0.0771	43	3.03e-04	4.26e+02	99.39	9.79e+03	0.00
18	1.00e-10	1.00e-02	0.7729	0.0644	30	7.97e-03	1.65e-02	99.39	6.92e+03	0.45
19	1.00e-10	1.00e-01	0.7568	0.0489	30	6.36e-03	2.07e-02	99.39	6.62e+03	0.02
20	1.00e-10	1.00e+00	0.8107	0.0568	31	3.50e-03	5.85e-02	99.39	6.61e+03	0.00
21	1.00e-10	1.00e+02	0.9441	0.0557	38	1.81e-03	4.27e+00	99.39	6.61e+03	0.00
22	1.00e-10	1.00e+03	1.0690	0.0761	42	8.64e-04	4.26e+01	99.39	6.61e+03	0.00
23	1.00e-10	1.00e+04	1.1615	0.0745	46	2.94e-04	4.26e+02	99.39	6.61e+03	0.00
24	1.00e-08	1.00e-02	0.7418	0.0569	29	6.30e-03	4.28e-01	99.39	4.05e+03	8.68
25	1.00e-08	1.00e-01	0.7346	0.0565	29	3.91e-03	5.54e-01	99.39	2.31e+03	2.23
26	1.00e-08	1.00e+00	0.7877	0.0502	31	2.72e-03	6.05e-01	99.39	2.00e+03	0.16
27	1.00e-08	1.00e+02	0.9932	0.0688	39	1.17e-04	4.78e+00	99.39	1.94e+03	0.00
28	1.00e-08	1.00e+03	1.0864	0.0768	43	1.65e-05	4.31e+01	99.39	1.94e+03	0.00
29	1.00e-08	1.00e+04	1.1854	0.0781	47	1.27e-05	4.25e+02	99.39	1.94e+03	0.00
30	1.00e-06	1.00e-02	0.7297	0.0537	28	7.33e-03	2.63e+00	99.39	3.41e+03	15.73
31	1.00e-06	1.00e-01	0.7271	0.0468	29	6.28e-03	3.61e+00	99.39	2.22e+02	6.32
32	1.00e-06	1.00e+00	0.7979	0.0538	31	1.66e-03	3.96e+00	99.39	1.33e+02	1.11
33	1.00e-06	1.00e+02	1.0200	0.0648	40	1.25e-05	7.91e+00	99.39	1.12e+02	0.01
34	1.00e-06	1.00e+03	1.0839	0.0566	42	1.14e-05	4.64e+01	99.39	1.12e+02	0.00
35	1.00e-06	1.00e+04	1.1118	0.0589	44	1.15e-05	4.28e+02	99.39	1.12e+02	0.00
36	1.00e-04	1.00e-02	0.7361	0.0505	28	7.67e-03	5.69e+01	99.39	3.45e+03	94.89
37	1.00e-04	1.00e-01	0.7186	0.0616	28	3.47e-03	8.09e+01	99.41	4.19e+01	20.13
38	1.00e-04	1.00e+00	0.7720	0.0515	31	9.86e-05	9.32e+01	99.41	2.81e+01	7.32
39	1.00e-04	1.00e+02	0.9400	0.0689	37	1.10e-05	1.16e+02	99.41	1.56e+01	0.07
40	1.00e-04	1.00e+03	0.9830	0.0572	39	1.20e-05	1.49e+02	99.41	1.53e+01	0.00
41	1.00e-04	1.00e+04	1.0460	0.0626	42	1.12e-05	5.22e+02	99.41	1.53e+01	0.00
42	1.00e-02	1.00e-02	0.7718	0.0551	30	8.24e-03	1.07e+03	99.39	3.45e+03	340.47
43	1.00e-02	1.00e-01	0.7283	0.0544	28	5.62e-03	2.65e+03	99.42	4.49e+01	170.83
44	1.00e-02	1.00e+00	0.7102	0.0636	28	2.38e-03	3.84e+03	99.44	2.85e+01	33.16
45	1.00e-02	1.00e+02	0.8035	0.0584	31	1.07e-05	4.47e+03	99.51	8.48e+00	1.33
46	1.00e-02	1.00e+03	0.8512	0.0570	33	9.47e-06	4.57e+03	99.53	5.13e+00	0.10
47	1.00e-02	1.00e+04	0.9005	0.0523	36	8.98e-06	4.79e+03	99.53	4.66e+00	0.00
48	1.00e+00	1.00e-02	0.8041	0.0554	31	9.93e-03	5.12e+03	99.40	3.40e+03	429.46
49	1.00e+00	1.00e-01	0.7579	0.0574	29	3.74e-03	2.11e+04	99.45	4.50e+01	300.62
50	1.00e+00	1.00e+00	0.7106	0.0632	27	2.16e-03	3.68e+04	99.54	2.81e+01	61.03
51	1.00e+00	1.00e+02	0.6850	0.0667	26	8.39e-06	3.97e+04	99.74	9.06e+00	2.11
52	1.00e+00	1.00e+03	0.7175	0.0487	28	2.64e-06	4.07e+04	99.91	1.60e+00	0.18
53	1.00e+00	1.00e+04	0.7876	0.0473	30	1.85e-06	4.10e+04	99.93	6.98e-01	0.00
54	1.00e+02	1.00e-02	0.8186	0.0572	31	2.55e-02	8.10e+03	99.40	3.38e+03	436.50
55	1.00e+02	1.00e-01	0.7745	0.1278	30	9.68e-03	4.77e+04	99.46	4.51e+01	318.53
56	1.00e+02	1.00e+00	0.7261	0.0879	28	3.57e-03	7.84e+04	99.59	2.84e+01	63.02
57	1.00e+02	1.00e+02	0.6717	0.0765	26	1.24e-04	8.19e+04	99.80	9.01e+00	2.11
58	1.00e+02	1.00e+03	0.6802	0.0809	27	9.43e-06	8.23e+04	99.97	1.05e+00	0.18
59	1.00e+02	1.00e+04	0.7470	0.0612	29	7.70e-07	8.23e+04	100.00	2.91e-02	0.00
60	1.00e+04	1.00e-02	0.8041	0.0631	31	1.52e-01	8.12e+03	99.40	3.38e+03	436.59
61	1.00e+04	1.00e-01	0.8187	0.0821	31	1.13e-01	5.25e+04	99.46	4.51e+01	319.26
62	1.00e+04	1.00e+00	0.7593	0.0829	30	5.31e-02	8.52e+04	99.60	2.84e+01	63.14
63	1.00e+04	1.00e+02	0.6940	0.0685	26	3.02e-03	8.85e+04	99.80	9.01e+00	2.11
64	1.00e+04	1.00e+03	0.7052	0.0607	27	4.36e-04	8.87e+04	99.98	1.03e+00	0.18
65	1.00e+04	1.00e+04	0.7463	0.0609	29	5.03e-05	8.87e+04	100.00	3.80e-04	0.00
66	1.00e+06	1.00e-02	0.7945	0.0560	31	8.24e-01	8.12e+03	99.40	3.38e+03	436.59
67	1.00e+06	1.00e-01	0.8174	0.0674	31	2.33e+00	5.25e+04	99.46	4.51e+01	319.28
68	1.00e+06	1.00e+00	0.7610	0.0746	30	1.73e+00	8.52e+04	99.60	2.84e+01	63.15
69	1.00e+06	1.00e+02	0.7135	0.0707	28	4.86e-02	8.86e+04	99.80	9.01e+00	2.11
70	1.00e+06	1.00e+03	0.7243	0.0656	28	7.77e-03	8.87e+04	99.98	1.03e+00	0.18
71	1.00e+06	1.00e+04	0.7623	0.0454	29	1.95e-03	8.87e+04	100.00	3.81e-06	0.00
72	1.00e+08	1.00e-02	0.7840	0.0544	30	3.20e+00	8.12e+03	99.40	3.38e+03	436.59
73	1.00e+08	1.00e-01	0.8048	0.0555	31	2.61e+01	5.25e+04	99.46	4.51e+01	319.28
74	1.00e+08	1.00e+00	0.7557	0.0708	30	4.02e+01	8.52e+04	99.60	2.84e+01	63.15
75	1.00e+08	1.00e+02	0.7521	0.0735	29	8.79e-01	8.86e+04	99.80	9.01e+00	2.11
76	1.00e+08	1.00e+03	0.7405	0.0486	29	1.82e-01	8.87e+04	99.98	1.03e+00	0.18
77	1.00e+08	1.00e+04	0.7882	0.0599	30	1.10e-01	8.87e+04	100.00	3.81e-08	0.00
78	1.00e+10	1.00e-02	0.7565	0.0555	30	1.85e+01	8.13e+03	99.40	3.38e+03	436.59
79	1.00e+10	1.00e-01	0.7855	0.0726	30	2.98e+02	5.28e+04	99.46	4.51e+01	319.28
80	1.00e+10	1.00e+00	0.7404	0.0743	29	8.28e+02	8.60e+04	99.60	2.84e+01	63.16
81	1.00e+10	1.00e+02	0.7325	0.0749	29	1.88e+01	8.86e+04	99.80	9.01e+00	2.11
82	1.00e+10	1.00e+03	0.7425	0.0705	29	1.05e+01	8.87e+04	99.98	1.03e+00	0.18
83	1.00e+10	1.00e+04	0.7912	0.0575	31	9.58e+00	8.87e+04	100.00	3.81e-10	0.00