Flow Trading

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

August 15, 2021
Preliminary Version

Abstract

We propose a new market design for trading financial assets. Our design combines three elements: (1) Traders submit persistent piecewise linear flow demand curves to trade in shares per second (Kyle and Lee, 2017). (2) Frequent batch auctions are held at regular intervals, such as once per second (Budish, Cramton, and Shim, 2015). (3) Traders may submit orders to trade portfolios of assets as if they were one asset. Market clearing quantities and prices are the solution to a quadratic program with linear constraints, constructed by attributing preferences to orders and maximizing imputed gains from trade. Clearing prices exist and define traded quantities uniquely. Calculating prices is computationally feasible. The market design corrects flaws in existing markets. It promotes economic efficiency by eliminating the arms race for speed and by reducing messaging costs. It fosters trust and confidence by improving transparency and simplifying participation. All executable orders trade at the same prices at the same time. There is no need for a time priority rule to ration executions at the best bid or best offer. Our contribution combines three elements: (1) a market design that implements trading portfolios, (2) a simple language for preference expression, and (3) existence, uniqueness, and numerical results to establish proof-of-concept. Our approach facilitates liquidity provision across time and across assets.

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

[†]Professor of Economics, University of Cologne and University of Maryland. He consults on market design and is an academic advisor to Carta on the design of a private equity exchange. Funded in part from the German Research Foundation (DFG, EXC 2126/1-390838866) and the European Research Council (grant 741409).

[‡]Professor of Finance, University of Maryland. He 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.

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

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

[‖]Other than government grants and university support, none of the authors has received funding to support the research described here.

1 Introduction

Description of the current design Current exchanges for trading equities 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 $124.11 per share or better," which has one maximum quantity and one limit price. The orders are processed continuously, one-at-a-time in sequential order, with incoming "executable" orders matched in whole or in part with "nonexecutable" orders resting in the limit order book. Orders are typically for single securities rather than portfolios of securities. Displayed bids and offers respect a minimum "tick size," which is typically $0.01 per share for U.S. stocks, and minimum "lot size," which has historically been 100 shares for most U.S. stocks. In some markets, traded quantities must also respect a minimum tick size and minimum lot size.

This market design makes it costly for investors to implement trading strategies. Since the entire quantity for an order resting in the limit order book is subject to immediate execution against the next incoming executable order, the resting order risks being "picked off" or "sniped" by a high-frequency trader acting on new information such as new bids, offers, or trades in the same or related securities. On exchanges which process older resting limit orders before new ones at the same price, the discrete minimum tick size, such as $0.01 per share, induces a race for "time priority" to be the first in the limit order book. The value of being the first in the queue of orders at the best bid or offer increases when the minimum tick size is larger as a fraction of the stock price. A large minimum tick size thus imposes unnecessary bid-ask spread costs on a retail investor who places an immediately executable order for a small quantity.

Institutional investors often strategically choose to execute large quantities by smoothing their trading out over time to reduce price impact. Doing this efficiently requires placing and canceling hundreds or thousands of small orders for individual stocks. This requires institutional investors to have access to complex, expensive trading platforms to manage thousands of orders. Sophisticated investors often implement long-short arbitrage trades by buying one stock and selling another stock based on the difference between the price of one stock and a constant multiple of the price of another stock. Although such "pairs trades" are conceptually simple and easily described, placing orders for one stock at a time requires constantly canceling and replacing orders for both stocks as prices fluctuate. Implementing such strategies also requires access to complex, expensive trading platforms.

Flow trading To reduce the costs that the current equity exchanges impose on retail and institutional participants, we propose a new market design called "flow trading."

Flow trading is a combination of three key elements (Table 1). First, instead of placing orders

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

Table 1: Comparison of Flow Trading with Traditional Exchange

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

Second, instead of executing orders one at a time in sequence, orders are processed in frequent batch auctions held at fixed discrete intervals, such as once per second. Each order in the batch is treated the same regardless of when it arrived in the batch interval. The first order has no advantage over the last. Suppose batch auctions are held once per second. An order to buy at a maximum rate of one share per second will buy one share per batch if fully executable, a fraction of a share per batch if partially executable, or no shares per batch if non-executable ("off the market"). Prices and quantities are approximately continuous. Tiny fractions of shares can trade each second within a nearly continuous price grid. Orders persist over many auctions; execution stops when a user-defined termination criterion is met, such as the cumulative purchase of 1000 shares.

Third, instead of orders for a single asset, each order is for a portfolio of assets. A "portfolio" is a user-defined linear combination of assets in which the asset weights can be positive or negative. Portfolio orders allow assets to be either complements or substitutes. If two assets 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, such as a pairs trade, then the assets are substitutes because an increase in the price of one asset increases the quantity demanded of the other. For example, a pairs trade has a positive weight on the stock being bought and a negative weight on the stock being sold. An order to sell the S&P 500 has negative weights on each of the 500 stocks in the index. A standard limit order has a nonzero weight only on the stock being bought or sold. An order to sell a single asset, which represents an upward-sloping supply curve for the asset, is

implemented as an equivalent downward sloping demand curve for a portfolio with a negative weight on the asset sold and zero weight on other assets.

Benefits There are three main benefits. First, investors can directly express their preferences to the market. Investors can easily control the urgency of trade by choosing the maximum flow rate. Since trading fast incurs large temporary market impact, some traders may decide to trade more gradually, while those with short-lived information may still want to trade more quickly. Investors can also submit one order to obtain their optimal portfolio by specifying all assets and their respective weights. For example, they can effectively construct and customize their own index ETFs. Further, investors can use portfolio orders to execute a "Buy A, Sell B" pair trade strategy. When used for two highly correlated assets, like A shares and B shares of Berkshire Hathaway, it can prevent the prices from ever diverging in the first place, like a string that ties prices together.

Second, the new market design improves transparency and fairness. It is fair because all orders that arrive for the same batch auction are treated the same way. All quantities eligible for execution are executed, and all orders for the same asset receive the same price. It is transparent because, for example, a retail investor who trades 100 shares over a minute will be able to infer the execution rate on his order from publicly announced market clearing prices exactly. By this metric, traditional market designs are unfair. Due to discrete tick size, the executable quantity on an arriving order is frequently less than the total resting bid or offer quantities the order matches against. Some orders are selected over others, and investors cannot infer whether their orders are executed and why from publicly available information.

Third, the market design reduces the advantage of speed in trading technology and the risk of regular traders' resting orders being sniped by faster traders. Since all orders arriving for the same batch are treated the same, the value of speed and the incentive to become faster are both reduced. Further, flow trading makes the executed quantity proportional to the length of time. Even when new information arrives just before the next batch auction so that regular traders suffer sniping, competition among fast traders will quickly adjust prices, and the actual quantity executed at unfavorable prices will remain small.

Technical Foundations Establishing the effectiveness of our approach requires showing that clearing prices and quantities exist, are unique, and can be calculated quickly.

To find equilibrium quantities, we transform the problem into a well-understood quadratic optimization problem with linear constraints. To do so, we first formulate a quasilinear quadratic utility function by interpreting each order as an expression of preferences defining a linear marginal utility curve over the range where it is partially executable. Then the restriction that

the order must execute at a rate between zero and its maximum rate, say one share per second, is a linear inequality constraint, and market clearing defines a linear equality constraint for each asset. Since zero trade is feasible, the quadratic program yields unique equilibrium quantities.

Equilibrium prices are found as Lagrange multipliers of the dual problem. Regardless of whether assets are complements or substitutes, prices exist because downward slopping demand curves portfolios imply a convex objective function. Prices, however, may be nonunique 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 nonexecutable 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. This can also be done by the exchange acting as a backup market maker willing to buy or sell small quantities of each asset at any price.

Summary of computation here.

Other issues 1. possible applications to other markets 2. come back here after sec 5 3. policy advantages hard to quantify

1.1 Related Literature

The key ideas motivating our market design proposal— piecewise linear demand schedules, portfolios as linear combinations of assets, general equilibrium theory, quadratic programming, reducing temporary price impact by trading slowly—are well-understood by researchers in economics and finance. Our contribution is to show that these ingredients can be combined into a coherent and practical design for trading divisible assets like stocks, bonds, and futures contracts. Our proposed market design creates a simple language that institutional and retail traders can use to implement the complex strategies seen in today's markets.

Our flow trading proposal is a combination of several familiar ideas. 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 to deter an economically wasteful arms race for speed. The idea that quantities demanded are functions of prices for all assets is familiar to economists, and the idea that portfolios of assets rather than individual assets should be optimized is familiar to finance researchers. A related market design already exists in wholesale electricity markets (Cramton, 2017). Participants submit piecewise-linear curves in periodic auctions for electricity flows. The system operator determines clearing prices and quantities by solving a quadratic optimization with linear constraints.

Sophisticated expression of preferences over multiple objects is a theme in the market design literature (Lahaie and Parkes, 2004; Sandholm and Boutilier, 2006; Cramton, 2017). Bichler (2017), Parkes and Seuken (2018), and Vohra (2011). Budish, Cachon, Kessler, and Othman (2017), Budish and Kessler (forthcoming) describe previous research by computer scientists, economists, and operation researchers. Preference elicitation and the promotion of trust and privacy are addressed by Parkes, Rabin, Shieber, and Thorpe (2008); Parkes, Rabin, and Thorpe (2009); Parkes, Thorpe, and Li (2015); Thorpe and Parkes (2007, 2009). Our approach allows simple and powerful expression of preferences for trading tailored to the manner in which traders use markets for financial assets.

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, Du and Zhu (2017) and Vayanos (1999) 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. Li, Wang, and Ye (2021) model how the value of being the first in the queue of orders at the best bid or offer increases when the minimum tick size is larger as a fraction of the stock price.

Our approach to market design attempts to improve the operational efficiency of markets by lowering the costs of implementing the strategies that traders choose to implement in practice under the traditional market design. This goal of improving operational efficiency is different from the alternative goal of improving economic efficiency by mitigating the effects of externalities associated with exercise of market power or trading on private information. This alternative approach is an interesting subject of recent research. In the context of specific one-period-model of hedging, Rostek and Yoon (2020*a*,*b*) and Wittwer (2021) show that it is sometimes possible to improve welfare by requiring traders to trade assets separately rather than trade portfolios of assets simultaneously. In the context of a different one-period model with exponential rather than normal distributions, Duffie and Zhu (2017) show that a market design based on "size discovery"—a mechanism in which an auctioneer announces prices and traders indicate quantities they are willing to trade at the specified price—can improve welfare by mitigating traders' strategic incentives to reduce trading costs by shading quantities in Walrasian auction markets. Zhang (2020) proposes a market design that mitigates bid-shading by a for-

mula that taxes traders who take liquidity and subsidizes traders who provide liquidity. Traders are rewarded for placing orders close to the market, even if they are not executed. There is also an older literature that proposes "sunshine trading," which we can define as single-price auctions in which buyers or sellers seek to avoid imposing adverse selection on others by transparently announcing quantities before the auction is held (Wunsch, 1986). It is not clear how well these ideas generalize from a one-period model to a dynamic model with many rounds of trading.

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") use fixed-point theorems to derive computationally impractical theoretical results for preferences which include income effects. By contrast, our market design approach focuses on a language for preferences that yields existence and uniqueness within a computationally tractable framework. Thus, we exploit the theory of convex optimization for our results.

There are three main differences with 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.[1] 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 quasilinear utility functions of the form $u(\mathbf{x}) - \boldsymbol{\pi}^{\mathsf{T}}\mathbf{x}$, 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 quasilinear utility, which makes the second derivative of the utility

[1] There are two steps in our approach. In the first step, each trader chooses his optimal strategy, which is a downward-slopping demand curve for a portfolio. The portfolio determines the assets that the trader wants to buy or sell and the relative weights among them. In the second step, we determine the equilibrium that clears the market for all assets jointly. We do not specifically consider the first step and take as given the demand curves and the portfolios that the traders choose.

function negative (semi) definite, and this guarantees that all equilibrium must lie in a convex set. In our framework, substitutes and complements do not matter for existence or uniqueness, since the matrix is negative semidefinite anyway, but substitutes and complements matter a great deal for computational performance.

Indivisible Goods. Our assumptions are in some respects more similar to assumptions made in the literature on indivisible goods, which typically uses quasilinear utility.

The recent literature on indivisible goods tries to generalize the concept of substitutes from divisible assets to indivisible assets by generalizing various ways of thinking about substitutes. Hatfield et al. (2019) show that many different ways of defining substitutes with indivisible assets all are equivalent. Baldwin and Klemperer (2019), on the other hand, use tropical geometry to show that existence can be obtained not only when goods are substitutes but also in some cases when they are complements. Perhaps their paper can be understood as a way of generalizing the concept of substitutes from divisible goods to indivisible goods while recognizing that substitutes is not the most general assumption needed to obtain existence. Substitutes may be helpful, but not necessary, for obtaining existence.[2]

Structure of the paper The rest of the paper is structured as follows. Section 2 describes flow trading portfolio orders. Section 3 discusses the existence and uniqueness of market clearing prices and quantities. Section 4 provides a characterization of equilibrium 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 Trading Portfolio Orders

Formal Definition of Flow Orders Representing orders requires a language with which traders can express in mathematical form a demand curve to trade as a flow the assets in a user-defined portfolio. Demand curves are piecewise linear functions that define the rate at which portfolio assets are bought as a function of the price of the portfolio. Portfolios are linear combinations of assets, with positive weights on assets being bought and negative weights on assets being sold. The graph of a demand curve may have several different downward-sloping linear segments. We define an "order" as a demand curve with only one downward-sloping seg-

[2]Assets can be rotated (by invertible linear transformation) without changing the underlying economics of the problem. Whether assets are substitutes or complements depends on how the rotated assets are defined. Rotating assets plays out differently in models with divisible versus indivisible assets. While there may be a natural rotation of assets in economics, there is not a strong natural rotation of assets in finance, as the literature on factor pricing suggests. For computation, it usually makes sense to rotate assets so that matrices are sparse.

ment. Arbitrary piecewise-linear demand curves are constructed as sums of orders. Arbitrary downward-sloping functions can be approximated with piecewise-linear functions.

An order is defined by two sets of instructions: "immediate" instructions, which describe the demand schedule submitted to the next auction, and "interbatch" instructions, which describe how the order participates in future auctions.

The immediate instructions for the ith order contain exactly four elements: (1) a vector[3] of portfolio weights $\mathbf{w}_i$, which defines the portfolio being traded; (2) the maximum trade rate q_i; (3) the upper limit price p_i^H, which defines the upper end of the downward-sloping segment where the order begins to execute, and (4) the lower limit price p_i^L, which defines the lower end of the downward-sloping line segment where the order is fully executable. The interbatch instructions described the maximum cumulative quantity to be bought over many auctions, a cancellation time, or more complicated contingencies such as stop-loss conditions.

We describe the details next. Intuitively, a portfolio is a list of assets with positive or negative weights for each asset. Suppose the tradeable universe consists of N assets with say $N = 5000$. Portfolio weights are represented as a vector of length N. Mathematically, the portfolio weight vector $\mathbf{w}_i$ has the form

$$\mathbf{w}_i := (w_{i1}, \ldots, w_{iN})^\top, \qquad \text{where} \qquad w_{in} \in \mathbb{R}, \text{ for all } n. \tag{1}$$

Here, $i = 1, \ldots, I$ identifies the order, $n = 1, \ldots, N$ identifies the assets in the market, and $\mathbf{w}_{in}$ defines the ith order's portfolio weight on asset n as the number of shares of asset n in one unit of the portfolio.

The market price for a portfolio is the weighted sum obtained using market prices and nonzero portfolio weights. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ denote the column vector of market prices of all assets $n = 1, \ldots, N$. The market price of the portfolio is the inner product

$$p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi} := \sum_{n=1}^{N} w_{in} \pi_n. \tag{2}$$

A flow order for a portfolio $\mathbf{w}_i$ defines the flow demand $D_i(p_i)$ as a piecewise linear function of the portfolio price $p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi}$ with one downward-sloping segment in the following way:

$$D_i\left(p_i \,\middle|\, \mathbf{w}_i, q, p_i^L, p_i^H\right) = q_i \,\mathrm{trunc}\left(\frac{p_i^H - p_i}{p_i^H - p_i^L}\right), \qquad \text{where} \qquad \mathrm{trunc}(z) := \begin{cases} 1, & \text{for } z \geq 1 \\ z, & \text{for } 0 < z < 1 \\ 0, & \text{for } z \leq 0 \end{cases}. \tag{3}$$

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

The equation above does not depend on $\mathbf{w}_i$ because it defines the number of portfolio units demanded. The quantity q_i is a positive number defining the maximum rate at which the trader buys units of the portfolio. It defines the urgency of order execution.

The actual buy rate depends on the portfolio price as well as the upper and lower limit prices p_i^H and p_i^L. If the portfolio price p_i is less than or equal to the lower limit price 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 the upper limit price p_i^H, then the order is "nonexecutable" and ceases to 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 limit prices. When the order is partially executable, the relationship between prices and quantities is one-to-one; otherwise, it is many-to-one.

Remarks This approach not only generates compact notation for representing both buying and selling but also emphasizes a symmetry between buying and selling which will be important for understanding how market clearing works. General equilibrium theory often uses this idea that an upward sloping supply curve for positive quantities is equivalent to a downward sloping demand curve for negative quantities.

Although we think of "orders to buy portfolios," it is actually the assets in the portfolio which are bought or sold proportionally to their portfolio weights. For example, a pairs trade order might be filled by matching against two orders for the single assets, a buy order for the asset the pairs trade sells and a sell order for the asset the pairs trade buys. Since the market clears for assets, not portfolios, the maximum rate at which the trader buys individual assets is represented by the vector $q_i\mathbf{w}_i$, where the rate at which order i buys individual asset n is $q_i w_{in}$.

Equation (3) expresses the rate at which portfolio units are bought as a function of the four pieces of current order information: the portfolio weights $\mathbf{w}_i$, the maximum execution rate q_i, the upper portfolio price limit p_i^H, and the lower price limit p_i^L. The standard kind of interbatch information is the maximum cumulative portfolio quantity to be bought, denoted $Q_i^{\max}$. This interbatch information is also part of the specification of the order, but it is not shown in equation (3) because it usually does not affect the execution rate at a specific batch auction. In general, an order lives for multiple batch auctions. It continues to execute over multiple consecutive auctions at a rate between zero and q_i until the quantity $Q_i^{\max}$ is reached. The execution rate fluctuates over time as prices fluctuate between the upper price limit p_i^H and the lower limit p_i^L. As the order is executed, the exchange keeps track of the cumulative executed quantity and the remaining unexecuted balance on the order. Execution stops when the $Q_i^{\max}$ has been bought. Even though interbatch order information does not change the current order terms at most auctions, it does change the terms when the contingencies become binding con-

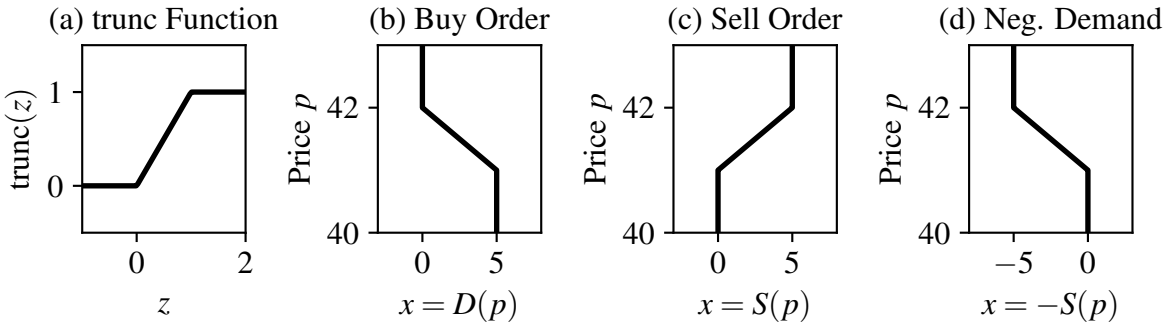


Figure 1: Plots of (a) the function trunc(z), (b) a single buy order $x = D(p)$ with one downward-sloping linear segment, (c) a single sell order $x = S(p)$ plotted as an upward-sloping supply curve with one upward-loping linear segment, and (d) the same sell order plotted as downward-sloping demand for negative quantities. The figures for buy and sell orders are plotted with flow quantity on the horizontal axis and price on the vertical axis.

straints. For example, when the order is almost fully executed, the order needs to be adjusted when the cumulative quantity filled is so close to $Q_i^{\max}$ that $Q_i^{\max}$ would be exceeded at the next auction under full execution at current terms.[4]

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

Illustrations Figure 1(a) shows the function trunc(z), which is identical to the cumulative distribution function for a uniform distribution on the interval $[0,1]$.[5]

Figure 1(b) shows the inverse demand function defined by an order for a single stock with upper limit price p_i^H = \$42.00, lower limit price p_i^L = \$41.00, and maximum execution rate

[4]To simplify exposition, we ignore the following detail. If the unexecuted balance on the order is so small that executing at the maximum rate would overfill the order by buying a cumulative number of portfolio units larger than $Q_i^{\max}$, then the lower price limit p_i^L is adjusted upward by the minimum amount required to avoid overfilling the order. This adjustments fills the order exactly at the price which fills the unmodified order and also gives the same executed quantity and price when the unadjusted order would be partially filled. Adjusting the lower limit price upward is better than adjusting the upper limit price downward or adjusting the maximum execution rate downward because such adjustments might result in underfilling the order when the unadjusted current terms would have filled it completely. Adjustments like this illustrate how interbatch instructions affect order execution.

[5]More generally, it would be possible to replace the function trunc(z) with an arbitrary cumulative distribution function. A smoother replacement function would make the demand function smoother and perhaps lead to more stable numerical properties. A function which never achieves the lower bound value of zero would define an order for which zero trade is not feasible. This would make proving existence of equilibrium more difficult.

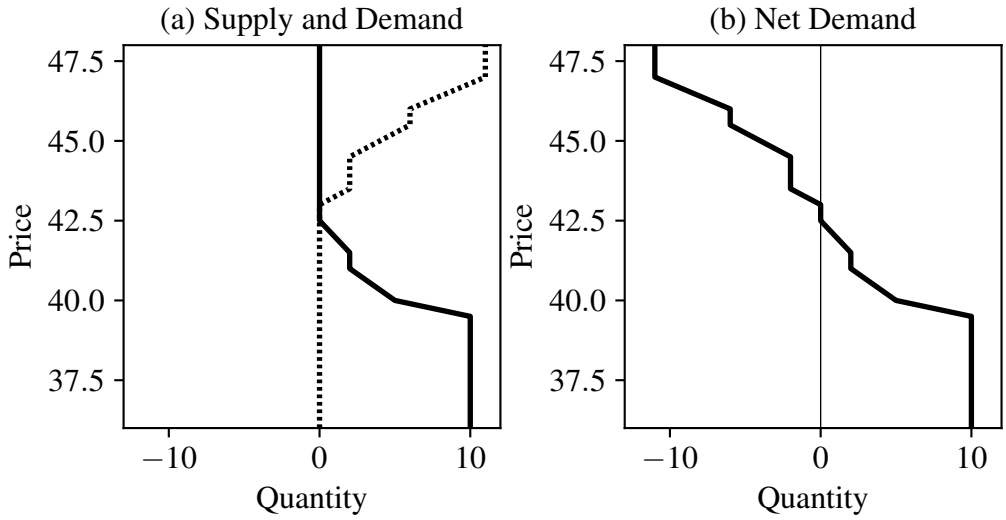


Figure 2: Plots of (a) three buy orders (solid) summed together and three sell order (dashed) summed together, (b) the same buy and sell orders presented as a downward-sloping net demand curve for both positive quantities (buy orders) and negative quantities (sell orders). The orders can be summed together without generating trade because the lowest limit price on the buy orders is less than the highest limit price on the sell orders.

$q_i = 5.00$ shares per second. The strictly downward sloping segment defines the region of partial execution, where the relationship between prices and quantities is one-to-one. The vertical segment on the price axis is the region of no execution, and the vertical segment through q_i is the region of full execution. On both vertical segments, quantities can be inferred from prices, but prices cannot be inferred from quantities because the relationship between prices and quantities is many-to-one.

Figure 1(c) shows an order to sell a single asset with the same price limits as the buy order and same maximum sell rate $q_i = 5.00$ shares per second. Figure 1(d) shows the same order as an economically equivalent downward-sloping demand curve for negative quantities.

Figure 2(a) shows a downward-sloping piecewise-linear demand curve and a piecewise linear supply curve (dashed). Since the demand curve has three distinct downward-sloping segments and the supply curve has three distinct upward-sloping segments, each curve can be defined by combining three orders. The second panel of Figure 2(b) shows a net demand curve in which the supply curve has been implemented as a demand curve to buy negative quantities and added to the demand curve. Since the upper limit on each buy order is below the lower limit on each sell order, these orders do not create trading volume by executing against one another. All six might all be submitted by the same trader as a market-making strategy, which buys at low prices and sells at high prices.

Examples A standard limit order expresses preferences to buy or sell a fixed quantity of one asset at one limit price. A flow order can be specified to approximate a standard limit order. First, when only one weight w_n is nonzero, the order is a simple order to buy one asset if the weight is positive or to sell one asset if the weight is negative. Second, the maximum rate q_i can be set to equal the quantity the trader wants to buy or sell, $Q_i^{\max}$ (say 1000 shares per second). If fully executable, this order would purchase 1000 shares at the next batch auction. Third, and lastly, the upper limit price p_i^H can be set to equal the price at which the trader wants to start buying or selling, and the lower limit price p_i^L can be set as closest as possible to p_i^H. Theoretically, we obtain a standard limit order in the limit $p_i^L \to p_i^H$. We, however, require the lower limit price to be strictly lower than the upper limit price, and the restriction, as we shall see, is critical to ensuring the unique equilibrium quantity allocations. For practical purposes, the approximated flow order will be essentially identical to the standard limit order.

Executing a pairs trade or long-short strategy, in general, can be done simply by selecting a portfolio with positive weights on assets being bought, negative weights on assets being sold, and zero weights on all other assets.

Traders can construct and trade their own index portfolios. For example, an order to buy or sell S&P 500 has positive weights on each stock in the S&P 500 index, with weights proportional to S&P 500 weights and zero weight on stocks not in the S&P 500 index. An order to sell an index has negative weights on all stocks in the index. Traders can easily customize the index portfolios by adjusting portfolio weights: overweighting desirable assets and underweighting the others.

Since trading at a fast rate can quickly move prices against the traders, traders will choose their maximum rate qi, trading off their urgency and temporary price impact. While some traders may choose to trade gradually at more favorable prices, other traders may still choose to trade rapidly to satisfy their urgent trading needs, such as taking advantage of their short-lived information.

Interbatch Instructions. Except for keeping track of the cumulative quantity executed, we ignore interbatch instructions, which apply across auctions. These instructions describe how a trader's orders evolve based on the history of prior outcomes. It likely makes sense for the exchange to include some additional common interbatch instructions. The exchange might allow orders with instructions to cancel at a specific time or to cancel when far away from being executable at current prices. In addition, the exchange might allow limit prices on orders to change automatically as a function of past prices to be closer to the market. This would facilitate market-making orders. The economic purpose of these orders is to reduce message costs between traders and the exchange by incorporating into one message with interbatch instructions a dynamic trading strategy. Many brokers and institutional traders will implement their

own dynamic strategies with a sequence of orders. Still, it is helpful if the standard dynamic strategies can be expressed as interbatch instructions. This is similar to the inclusion of proxy bids in multi-round auctions. The proxy bid provides an instruction that is relevant across multiple rounds of the auction. This enables bidders with simple strategies to participate in the auction more easily. They can place a single bid in round 1 and then ignore the dynamics. Portfolio limits to manage risk are a natural interbatch instruction. Indeed, the exchange is likely to implement portfolio limits as one component of risk management to prevent a trader from destabilizing the market with a fat-finger error or reckless trading strategy.

Stop-loss orders involve interbatch instructions to submit an order to buy or sell after when a prices rise above or fall below a target level.[6]

Algorithmic Frontiers. Possible extensions to flow trading involve instructions based on other traders' willingness to trade in the current auction. Flow trading handles TWAP orders automatically but does not handle VWAP orders. It may be possible to enrich the language to allow instructions like, "Execute this portfolio order only up to ten percent of trading volume at the current auction." Another extension is making order execution contingent on price impact. For example, "Execute this order only to the extent that its price impact is less than ten basis points." Such an extension is possible with additional computation, requiring a longer batch interval.

3 Market Clearing Prices and Quantities

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

3.1 Simple Illustrations

To convert the traders' demand curves for portfolios as a function of portfolio prices into a market demand curve for assets as a function of asset prices, first replace the portfolio price p_i by the price implied by asset prices, $p_i = \boldsymbol{\pi}^\mathsf{T}\mathbf{w}_i$, then convert the demand for portfolio units D_i into the demand for individual assets by multiplying by the portfolio weights $\mathbf{w}_i$. Summing up the demand for assets across orders yields a market net excess demand curve for assets as a function of asset prices:

$$D(\boldsymbol{\pi}) := \sum_{i=1}^{I} D_i\left(\boldsymbol{\pi}^\mathsf{T}\mathbf{w}_i \,\middle|\, \mathbf{w}_i, q_i, p_i^L, p_i^H\right)\mathbf{w}_i. \tag{4}$$

[6] Although stop-loss orders intuitively resemble demand schedules which slope the wrong way (upward), stop-loss orders are more accurately thought of as contingent orders are implemented immediately *after* a target price level is reached.

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

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

$$D(\boldsymbol{\pi}) = \mathbf{0}. \tag{5}$$

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

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

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

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

Figure 3 illustrates the simplicity of calculating clearing prices when all orders are for the same asset. Figure 3(a) shows a piecewise-linear demand curve and a piecewise-linear supply curve which intersect at a unique clearing price $p = \$45.00$ per share. The market trading volume is $x = 2.00$ shares per second. Figure 3(b) shows the combined demand and supply curve as one downward-sloping net demand curve, which has the same clearing price at a net quantity of zero. While the clearing price is easily calculated from the net demand curve, trading volume cannot generally be inferred from the net demand curve.

With many orders for different portfolios of multiple assets, calculating a clearing price is

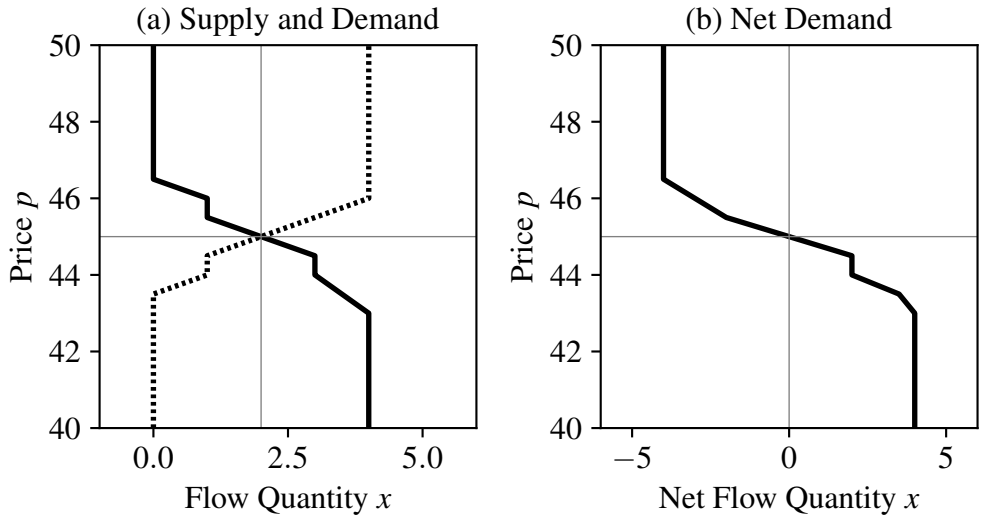


Figure 3: Plots of (a) a supply curve (dashed) intersecting a demand curve (solid) and (b) the net demand curve obtained by summing supply and demand. The supply and demand curves are each generated by summing together three orders. The clearing price of $45.00 per share generates trading volume of 2.00 shares per second. The clearing price and quantity can easily be calculated by bisection or trial and error.

mathematically more complicated because the intermediate value theorem and the bisection algorithm do not generalize in a natural way.

Example: Two Assets. Consider an example with two assets. Suppose there are only two orders in the market, a buy order for asset A and sell order for asset B. The only clearing prices involve no trade because neither order can find another order to trade with . The clearing prices which support no trade are not unique because any high enough prices for asset A and low enough prices for asset B will imply that both orders are non-executable. Add to this example a pairs-trade order for a portfolio with negative weight on asset A and positive weight on asset B, where upper and lower limit prices are set such the only clearing prices involve no trade. The set of no trade prices is the interior of the triangle defined by the three dotted lines in Figure 4. As indicated in grey, this region is the intersection of the three orders' halfplanes of no execution.

If the limits p_i^H and p_i^L on the pairs trade order order are changed, it is possible for trade to occur, as in Figure 5. Figure 5(a) shows the order for asset A as a solid line and plots the derived demand for asset A as a heavy dashed line. The derived demand takes into account the derived demand from the pairs-trade order by fixing the price of asset B at its market-clearing value. Figure 5(b) hows the order for asset B a solid line and the derived demand, holding the price of asset A constant at its market-clearing level, as a heavy dashed line. The pairs trade order makes the assets substitutes. Therefore, when the price of asset B is fixed at a level above its

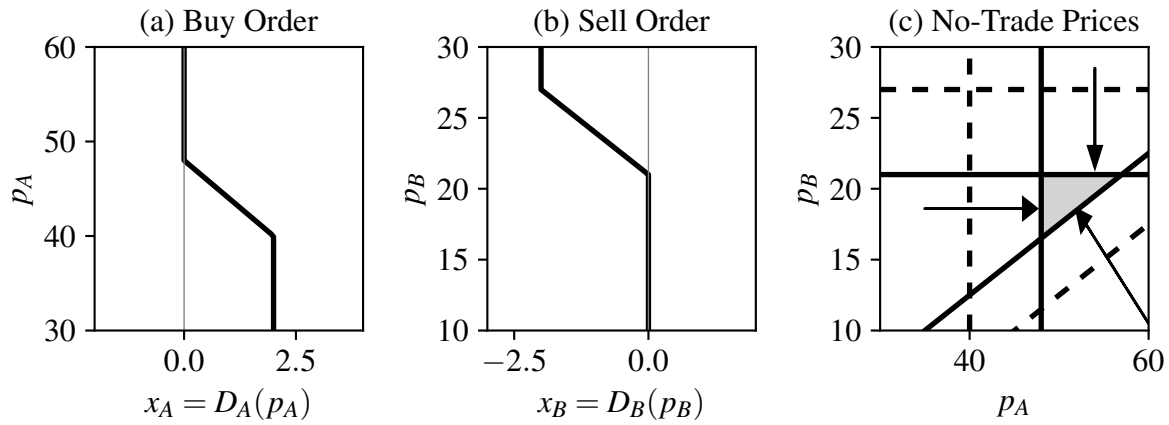


Figure 4: Plots of (a) a buy order for asset A; (b) a sell order for asset B; and (c) price pairs for the buy order, the sell order, and a pairs-trade order. In (c), the areas between the parallel solid and dashed lines indicate regions of partial execution of the orders. The solid lines bound the region of no execution and the dotted lines bound the region of full execution. The three orders generate no trade at any prices on the interior of the triangle defined by the intersection of the dashed lines. The arrows, which are proportional to the vectors of portfolio weights, indicate the direction in which the orders tend to push prices. In the Euclidian norm, the arrows are orthogonal to the solid lines they point to. As plotted, the arrows may not appear to be perpendicular to the lines they point to because one dollar on the vertical axis represents a different length than one dollar on the horizontal axis.

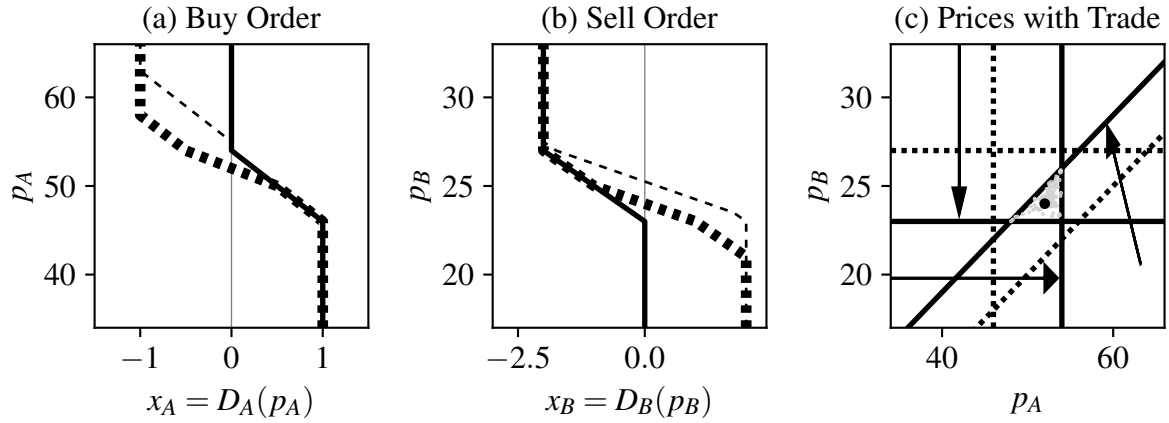


Figure 5: Plots of (a) a buy order for asset A, and sell order for asset B, and (c) price pairs for both assets. In (a) and (b), the solid line represents the order itself, the heavy dashed line represents the demand schedule generated by the order for the asset and the pairs-trade order fixing the price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

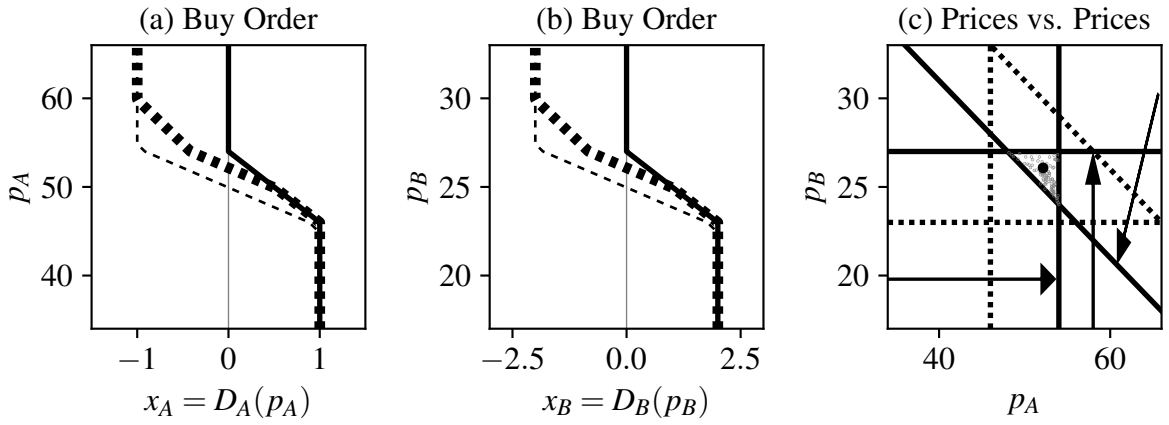


Figure 6: Plots of (a) a buy order for asset A and buy order for asset B, and (c) an index order to sell both assets. In (a) and (b), the solid line represents the order itself, the heavy dashed line represents the demand schedule generated by the order for the asset and the index order fixing the price of the other asset at its market-clearing level, and the light dash lines represents the demand for the asset from the order itself and the pairs-trade order fixing prices above the market clearing level for the other asset. In (c), the solid lines indicate boundaries between no execution and partial execution of the buy order, the sell order, and the index order. The dashed lines indicate boundaries between partial execution and full execution of the orders. Trade occurs at a price on the interior of the triangle defined by the intersection of the three solid lines. Arrows proportional to orders' portfolio weights indicate directions in which the orders push prices.

equilibrium price, the derived demand for asset A is shifted towards higher prices as well. The same is true for asset B. This is depicted in Figures 5(a) and 5(b) as the thin dashed lines, which for both asset have prices which are the same or higher than the thick dashed line. In Figure 5(c), the region where all three orders are partially or fully executable is the interior of the triangle defined by the intersection of the solid lines defining the boundary between no execution and partial execution of the orders.

Now, consider the case when there are buy orders for each asset A and B and a sell order for the index portfolio of assets A and B. Let the price limits on the index order be such that trade occurs only the the region where all three orders are partially executable. The portfolio order makes the assets complements. Therefore, when the price of asset B is fixed at a level above its equilibrium price, the derived demand for asset A is shifted towards lower prices. The same is true for asset B. This is depicted in Figures 6(a) and 6(b) as the thin dashed lines, which for both asset have prices which are the same or lower than the thick dashed line. The parallel lines defining the infinite strip in which the index order is partially executable are downward sloping rather than upward sloping because keeping the portfolio price constant requires as asset prices change requires increasing the price of one asset and decreasing the price of the other asset.

The examples with two assets in Figures 4, 5, 6 illustrate principles which generalize to mar-

kets with more than two assets. The parallel lines defining regions of partial execution become parallel hyperplanes. The arrows, which point in directions proportional to portfolio weights defined by the orders, become vectors perpendicular to the hyperplanes.

The regions where prices are not unique, such as regions of no trade in Figures 4, are convex sets which may either be bounded (compact) or extend to infinity in one or more directions. It is this property of the set of clearing prices that enables a simple tie-breaking rule to guarantee existence. For example, nearest-to-the-prior-price-vector identifies a unique price vector within the set of clearing prices, since the set of clearing prices is convex. Finding the nearest price vector is simply a secondary quadratic objective.

3.2 Existence of Clearing Prices and Quantities

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

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

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

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

Imputing utility functions to orders is a convenient mathematical modeling device. We proceed as though orders directly represent consumer preferences, even though, in practice, traders submit orders strategically. Thus, our methodology does not measure actual economic

welfare and does not generate welfare results on market efficiency. Still, the method provides a practical approach to prove that clearing prices and quantities exist.

Pseudo-Utility. Let $V_i(x)$ denote the dollar utility of order i from trading x portfolio units. To find $V_i(x)$, we first define the marginal utility function $M_i(x)$ as the inverse demand curve, which maps order i's execution rate $x \in [0, q_i]$ into prices $p \in [p_i^L, p_i^H]$, and is defined by

$$M_i(x) := p_i^H - \frac{p_i^H - p_i^L}{q_i} x \qquad \text{for } x \in [0, q_i]. \tag{6}$$

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

$$V_i(x) := \int_0^x M_i(u)\,\mathrm{d}u \tag{7}$$

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

$$V_i(x) = p_i^H x - \frac{p_i^H - p_i^L}{2q_i} x^2 \tag{8}$$

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

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

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

$$V(\mathbf{x}) := \sum_{i=1}^{I} V_i(x_i), \tag{9}$$

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

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

rate constraints:

$$\max_{\mathbf{x}} V(\mathbf{x}) \qquad \text{subject to} \begin{cases} \sum_{i=0}^{I} x_i \mathbf{w}_i = \mathbf{0} & \text{(market clearing)} \\ x_i \in [0, q_i] \text{ for all } i & \text{(order execution rate),} \end{cases} \tag{10}$$

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

Indeed, since the objective function is quadratic and the constraints are linear, this is a quadratic program (QP). To make this quadratic structure apparent using matrix and vector notation, let $\mathbf{W}$ denote the $N \times I$ matrix whose ith row is $\mathbf{w}_i{}^{\mathsf{T}}$. 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 (10) may be written compactly as

$$\max_{\mathbf{x}} \left[\mathbf{x}^{\mathsf{T}} \mathbf{p}^H - \tfrac{1}{2} \mathbf{x}^{\mathsf{T}} \mathbf{D} \mathbf{x} \right] \qquad \text{subject to} \qquad \mathbf{W}\mathbf{x} = \mathbf{0} \qquad \text{and} \qquad \mathbf{0} \le \mathbf{x} \le \mathbf{q}. \tag{11}$$

The solution of a QP involves solving a linear system. Bertsekas (2009, Example 5.3.1, Quadratic Programming Duality, pp. 170–171) discusses the solution to an essentially equivalent problem based on theorems equivalent to our Theorems 1–4, which we discuss next.

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* (11).

Proof. The problem has the following properties:

1. Compactness: The order constraints define a compact convex set as the Cartesian product of I intervals, $[0, q_1] \times \cdots \times [0, q_I]$. The market clearing conditions are a linear constraint which defines a hyperplane. The intersection of a compact set and hyperplane is compact. Thus, we can think of the domain of $V(\mathbf{x})$ as a compact set.

2. Feasibility: No trade ($\mathbf{x} = \mathbf{0}$) generates well-defined utility for each order ($V_i(0) = 0$), no-trade clears markets, and a zero execution rate 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 on the domain $x_i \in D_i = [0, q_i]$. Since V is the sum of V_i across i, the function V_i is concave on the domain $\mathscr{D} = D_1 \times D_2 \times \ldots \times D_I$, where the dimensionality of the domain space is equal to the number of orders I. This domain $\mathscr{D}$ is also compact. The constraints are linear on all real numbers, not just in the domain.

It is a well-known principle of convex analysis (Boyd, 2004; Bertsekas, 2009) that compactness, feasibility, and strict concavity imply existence of a unique maximizing vector $\mathbf{x}^*$. □

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

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

$$L(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := \mathbf{x}^\mathsf{T}\mathbf{p}^H - \tfrac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{D}\mathbf{x} - \boldsymbol{\pi}^\mathsf{T}\mathbf{W}\mathbf{x} + \boldsymbol{\mu}^\mathsf{T}\mathbf{x} + \boldsymbol{\lambda}^\mathsf{T}(\mathbf{q} - \mathbf{x}). \tag{12}$$

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

The dual objective function associated with the primal problem of maximizing aggregate utility (11), is then defined by

$$\hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := \max_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \text{for} \qquad \boldsymbol{\pi} \in \mathbb{R}^N, \qquad \boldsymbol{\mu} \geq \mathbf{0}, \qquad \boldsymbol{\lambda} \geq \mathbf{0}. \tag{13}$$

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

$$g^* := \inf_{\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}} \hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \text{subject to} \qquad \boldsymbol{\pi} \in \mathbb{R}^N, \qquad \boldsymbol{\mu} \geq \mathbf{0}, \qquad \boldsymbol{\lambda} \geq \mathbf{0}. \tag{14}$$

The dual problem (14) 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* (14)*. The solutions* $\mathbf{x}^*$ *and* $(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *are a primal-dual pair which satisfies the strict duality relationship*

$$g^* = V(\mathbf{x}^*). \tag{15}$$

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

1. Concavity: The objective function $V(\mathbf{x})$ is 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. Feasibility: No trade ($\mathbf{x} = \mathbf{0}$) is feasible because (a) it lies (trivially) on the interior of the domain of V, which is all of $\mathbb{R}^N$; (b) satisfies the market clearing constraint; and (c) and satisfies the order execution rate constraints because a flow rate of zero (nonexecutable order) is allowed on every order.

4. 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.

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 (2004), Bertsekas (2009, Proposition 5.3.4, p. 173). Since Theorem 1 guarantees that a solution to the primal problem does exist, the solution to the primal primal has the same value as the solution to the dual problem. □

It follows from the market-clearing equality constraint that the solution $\boldsymbol{\pi}$ to the Lagrangian is a set of clearing prices. Another way to see this is that if the market were not to clear, $\boldsymbol{\pi}$ can be tweaked slightly to decrease $\hat{G}$.[8] Since the constraints on the order are inequality constraints, each of the individual multipliers must be nonnegative ($\mu_i \geq 0$ and $\lambda_i \geq 0$ for all i).

Theorem 2 does not guarantee that clearing prices are unique. The set of clearing prices is convex and may be unbounded, as in Figure 4 above. A trivial example occurs when all orders are buy orders for individual assets, and there are no sell orders. Then any sufficiently high price clears the market with zero trade. There may also be cases where the clearing price is not unique even when trade occurs. A trivial example occurs when there is one fully executable buy order and one fully executable sell order for the same asset, with the same quantities q and the lower limit price on the buy order strictly above the upper limit price on the sell order. We discuss a tie-breaking rule to pick a unique price in the next section.

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

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

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

4 Equilibrium Characterization and Computation

So far, we have shown that clearing prices exist and imply unique quantities, but we have not discussed whether it is feasible for an exchange to calculate these prices in a realistic setting. We discuss this issue in this section. In the first subsection, we discuss how the Karush–Kuhn–Tucker (KKT) conditions characterize a solution to the problem. In the next subsection, we discuss numerical algorithms for finding the solution.

4.1 Characterization of Equilibrium

The solution to both the primal problem and dual problem is characterized by the following theorem:

Definition 3. *The* Karush–Kuhn–Tucker (KKT) Conditions *for for primal feasibility, dual feasibility, primal optimality, and complementary slackness are*

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

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

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

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

Theorem 4 (Karush–Kuhn–Tucker (KKT) Conditions)**.** *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. There exists at least on solution to the KKT conditions. The solution for optimal quantities* $\mathbf{x}^*$ *is unique.*

Proof of Theorem 4. This is a straightforward consequence of Theorems 1 and 2, which imply that a unique optimal primal solution $\mathbf{x}^*$ exists and some optimal dual solution $(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ ex-

ists, and these solutions form a primal dual pair with the same optimized value; see Bertsekas (2009, Theorem 5.34(b), p. 173) .

Equation (18) presents first-order conditions for the primal optimality problem

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^I}{\operatorname{argmax}}\ L(\mathbf{x}, \boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \qquad \text{(Primal Optimality)}. \tag{20}$$

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

In a nutshell, the KKT conditions characterize optimal quantities, which lead to optimal prices. Alternatively, we can directly solve the KKT conditions for optimal prices, using the dual problem. This is attractive because there are likely to be far fewer prices and order quantities to solve for.

Solution to Dual Problem: Gains Function. The dual problem maps nicely into the way economists think about consumer surplus. This subsection spells out the details of the mapping.

The aggregate gains function is obtained from the dual problem by "minimizing out" the constraint-vector parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$:

Theorem 5 (Gains function)**.** *Define the gains function as*

$$G(\boldsymbol{\pi}) := \min_{\boldsymbol{\lambda},\boldsymbol{\mu}} \hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \textit{subject to} \qquad \boldsymbol{\lambda} \geq \mathbf{0}, \qquad \boldsymbol{\mu} \geq \mathbf{0}. \tag{21}$$

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

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi}\in\mathbb{R}^N}{\operatorname{argmin}}\ G(\boldsymbol{\pi}). \tag{22}$$

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

Proof of Theorem 5. It is clear from equation (12) that the order execution rate x_i and multipliers μ_i and λ_i may be chosen on an asset-by-asset basis, dramatically simplifying the problem, because the only connection between different orders operates through the price of the portfolio $p_i = \mathbf{w}_i{}^\top \boldsymbol{\pi}$. In the primal problem, x_i is chosen to maximize utility. In the dual problem, the exchange chooses prices $\boldsymbol{\pi}$ and multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ to minimize the utility, taking into account that x_i will be chosen optimally. Differentiating the Lagrangian with respect to λ_i and μ_i, it is clear that if x_i were to be negative (because the portfolio price is so high that the buy order would want to become a sell order in the absence of an explicit constraints $0 \leq x_i \leq q_i$), then

the exchange could reduce costs by increasing μ_i. Eventually, x_i is increased to the point where the constraint $x_i \geq 0$ is satisfied. Similarly, if the order would be overfilled due to low prices, the exchange would increase λ_i to lower costs, forcing the quantity x_i down to satisfy the constraint $x_i \leq q_i$. Once both order size constraints are satisfied, it is not optimal for the exchange to adjust μ_i or λ_i further because doing so would increase the objective being minimized. When the constraints are strictly satisfied, $0 < x_i < q_i$, the exchange chooses $\mu_i = \lambda_i = 0$; it could reduce costs by choosing $\mu_i < 0$ or $\lambda_i < 0$ but this would violate nonnegativity constraints on the multipliers. Altogether, the optimal μ_i and λ_i chosen by the exchange satisfy the complementary slackness conditions $\mu_i x_i = 0$ and $\lambda_i(q_i - x_i) = 0$ and x_i stays within the required bounds $0 \leq x_i \leq q_i$. Thus, the order i contributes to the Lagrangian by choosing x_i to maximize $V_i(x_i) - x_i \mathbf{w}_i{}^\mathsf{T} \boldsymbol{\pi}$ subject to $0 \leq x_i \leq q_i$. This is exactly the value of order i's gains function $G_i(x_i^*)$ in equation (24), defined as the value $V_i(x_i)$ minus the cost $x_i p_i$ with $p_i = \mathbf{w}_i{}^\mathsf{T} \boldsymbol{\pi}$. Thus, the Lagrangian is the sum of the gains functions $G_i(x_i^*)$, which is the desired result $G(\boldsymbol{\pi}) = \min_{\boldsymbol{\lambda},\boldsymbol{\mu}} \hat{G}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ subject to $\boldsymbol{\lambda} \geq \mathbf{0}$, $\boldsymbol{\mu} \geq \mathbf{0}$.

□

The economic interpretation of the gains function is that it measures the Marshallian consumer surplus obtained by trading at price p. For each order, the consumer surplus is defined as the integral of the difference between the marginal as-bid value and the price:

$$G_i(p) := \int_0^{D_i(p)} \big(M_i(z) - p\big)\, \mathrm{d}z, \tag{23}$$

which is the value function after subtracting the cost of buying the quantity executed:

$$G_i(p) = V_i\big(D_i(p)\big) - p \cdot D_i(p). \tag{24}$$

To provide intuition, Figure 7 illustrates the gains-from-trade for a buy order. The dollar gain is the area of a shaded region, which is a trapezoid when the order is fully executing ($p_i^H \leq p$), a triangle when the order is partially executing ($p_i^L < p < p_i^H$), and exactly zero when the order is not executing ($p_i^H \leq p$). In the region of partial execution, the gain is a quadratic function of p since the order itself if linear. In the regions of full execution and no execution, the gains function is linear in p because the order quantity is constant.

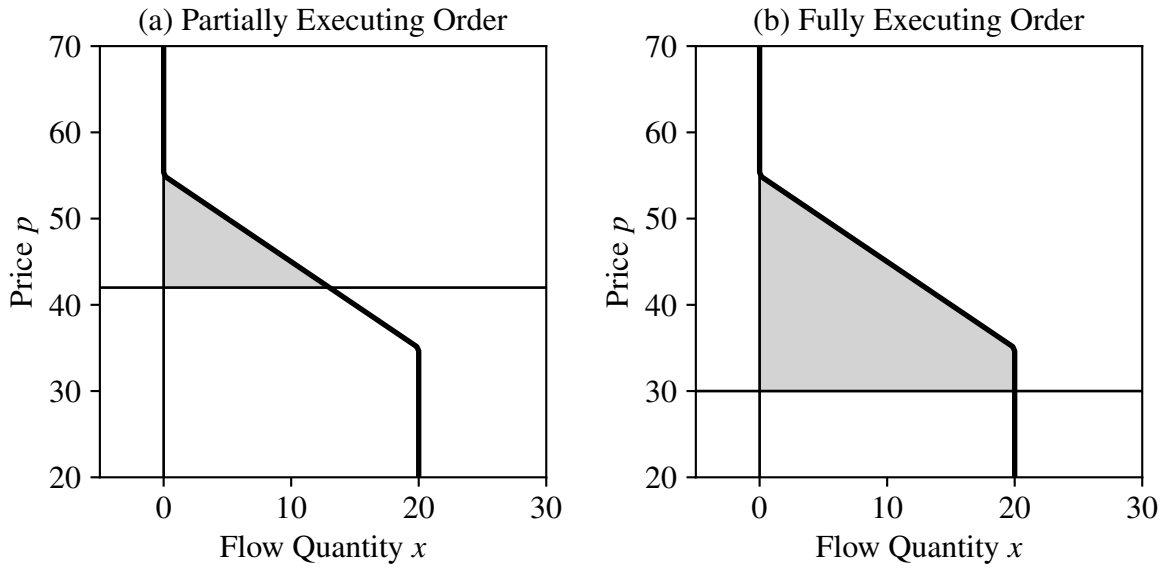


Figure 7: Plots of gains from trade for (a) a partially executing order and (b) and fully executing order. The gains from trade are represented by the area of the regions shaded in grey.

The explicit solution for $G_i(p)$ is

$$G_i\left(p \,\middle|\, \text{order}\right) := \begin{cases} q_i\left(\frac{1}{2}(p_i^H - p_i^L) - p\right) & \text{for } p < p_i^L \qquad \text{(fully executable)}, \\ \dfrac{q_i\,(p_i^H - p)^2}{2(p_i^H - p_i^L)} & \text{for } p_i^L \le p \le p_i^H \qquad \text{(partially executable)}, \\ 0 & \text{for } p_i^H \le p \qquad \text{(non-executable)}. \end{cases} \tag{25}$$

The three cases in equation (25) match up exactly with the three cases for the Lagrange multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$: (1) When the order is partially executable, the price of the portfolio $p_i^* := \boldsymbol{\pi}^{*\mathsf{T}}\mathbf{w}_i$ uniquely defines an execution rate $x_i^* \in (0, q_i)$ on the demand curve, and both multipliers μ_i^* and λ_i^* are zero. (2) When the ith order is non-executable, the portfolio price p_i^* is equal to or greater than p_i^H (the price below which the order becomes executable), the quantity traded x_i is zero, the multiplier λ_i^* is zero, and the multiplier μ_i^* measures the difference between the portfolio price p_i^* and p_i^H. (3) When the ith order is fully-executable order, the portfolio price p_i^* is less than or equal to p_i^L (the maximum price at which the order is fully executable), the quantity traded is q_i, the multiplier μ_i^* is zero, and the multiplier λ_i^* measures the difference between the portfolio price p_i^* and p_i^L.

Then the aggregate gains-from-trade function $G(\boldsymbol{\pi})$ is a function of the price vector $\boldsymbol{\pi}$ obtained by summing the individual gains-from-trade functions with the portfolio price calcu-

lated from the portfolio weights and price vector, $p_i = \boldsymbol{\pi}^\mathsf{T}\mathbf{w}_i$:

$$G(\boldsymbol{\pi}) = \sum_{i=1}^{I} G_i\left(\boldsymbol{\pi}^\mathsf{T}\mathbf{w}_i \,\middle|\, \mathbf{w}_i, q_i, p_i^H, p_i^L\right). \tag{26}$$

The objective function $G(\boldsymbol{\pi})$ is convex in $\boldsymbol{\pi}$. By construction, the gradient of the aggregate surplus function is minus the aggregate demand function: $\nabla G(\boldsymbol{\pi}) = -D(\boldsymbol{\pi})$. When the gradient is zero, markets clear and the aggregate gains function is maximized. When $G(\boldsymbol{\pi})$ is differentiable, the second derivative $\nabla^2 G(\boldsymbol{\pi})$ is an $N \times N$ matrix given by

$$\nabla^2 G(\boldsymbol{\pi}) = -\nabla D(\boldsymbol{\pi}) = -\sum_{i=1}^{I} \frac{\mathrm{d}D_i(p_i)}{\mathrm{d}p_i} \cdot \mathbf{w}_i\mathbf{w}_i{}^\mathsf{T}. \tag{27}$$

Thus, $\nabla^2 G(\boldsymbol{\pi})$ is the sum of positive semi-define matrices and therefore itself positive semidefinite. This makes $G(\boldsymbol{\pi})$ a convex function of $\boldsymbol{\pi}$. At points where G is not differentiable, convexity can be proven using subgradients.

4.2 Computational Feasibility

Focussing on the primal problem suggests solving for quantities. Focussing on the dual problem suggest solving for prices. The KKT equations can be written in a form which is symmetric in both quantities and prices. Therefore, the KKT equations themselves suggest solving for quantities and prices simultaneously. Does a good algorithm for calculating clearing prices focus on solving for quantities, for prices, or for both at the same time? Answering this question leads to a discussion of computational feasibility. Issues of computational feasibility are conceptually easier to discuss if the exchang itself is allowed to trade small quantities.

Exchange Trading. A practical algorithm for implementing flow trading must be able to calculate clearing prices in less than one second. The algorithm starts with a guess for clearing prices, then updates the guess at each iteration, eventually finding prices that reduce uncleared quantities to near zero. The algorithm terminates when uncleared quantities are less than some de minimus amount (say \$0.01 of uncleared dollar volume per asset) or when time runs out (after say 0.50 seconds).

To evaluate algorithms, it is useful to allow the exchange itself trades small quantities for its own account, like a market maker. The exchange conducts two kinds of trading. First, when the algorithm does not converge to a clearing price in the required time, the clearing price is defined as the algorithm's current best guess, and the exchange takes into its own inventory uncleared quantities. Such uncleared quantities are are likely small (say less than \$10.00 of dollar volume

per asset) but not de minimus. Second, the exchange places orders to trade for its own account. Exchange orders not only provide liquidity to other traders, but they also make it easier for the algorithm to find clearing prices. The exchange can liquidate the inventories it acquires by placing a portfolio order to liquidate its inventories gradually in subsequent auctions.

A discussion of computation must address three related computational issues:

1. Computational Feasibility: Successful implementation of flow trading requires a practical algorithm which can reliably calculate clearing prices in less than one second.

2. Backup Plan: In the rare cases when the algorithm cannot calculate clearing prices in less than one second, the market design requires a plan of action, such as a trading halt or trading delay.

3. Tiebreaking Rule: In cases where clearing prices are not uniquely defined, the algorithm must choose an official price from the set of clearing prices.

Allowing the exchange to trade addresses all three of these issues. Adding uncleared quantities to the exchange's inventory lowers the hurdles for computational feasibility and lessens the need to have a trading halt or a trading delay. For example, a brief trading delay might be needed only when the algorithm's best guess for clearing prices has uncleared quantities greater than some threshhold ($10.00 per asset). If the exchange has a partially executable order for each asset in that market at any price, the price associated with the quantity executed for the exchange's market making orders define a unique clearing price, making a tiebreaking rule unnecessary.

An effective algorithm makes the probability of invoking the backup plan very small. If the quantities traded by the exchange are sufficiently tiny, we can think of exchange trading as a way to discuss numerical convergence thresholds. If the quantities traded by the exchange are not economically negligible, then exchange trading becomes a substantial component of our proposed market design.

Mathematical Implementation of Exchange Orders. We propose that the exchange trade in the following manner. For each asset, the exchange defines a target market price. Let $\bar{\boldsymbol{\pi}}$ denote the vector of target prices. If markets clear at the target prices, the exchange does not trade.

The exchange trades quantities $\mathbf{Q}^{-1}(\boldsymbol{\pi} - \bar{\boldsymbol{\pi}})$, where $\mathbf{Q}$ is a diagonal matrix with strictly positive entries on the diagonal which functions of the asset's price, share volume, and volatility expected based on market data from the recent past. For each asset, the quantity traded is proportional to the corresponding diagonal element. The exchange therefore buys if the price $\boldsymbol{\pi}$ is below the target $\bar{\boldsymbol{\pi}}$ and sells if the price is above the target. A natural target price is the clearing price from the most recent auction.

These additional orders can be implemented by imputing a quadratic utility function to the exchange and adding this utility function to the optimization problem. The utility function has the form $\mathbf{y}^\mathsf{T}\bar{\boldsymbol{\pi}} - \frac{1}{2}\mathbf{y}^\mathsf{T}\mathbf{Q}\,\mathbf{y}$, where $\mathbf{y}$ is the vector of assets traded. More generally $\mathbf{Q}$ can be any $N \times N$ positive definite matrix, such as a positive linear combination of a diagonal matrix and a returns covariance matrix. Conceptually, the matrix $\mathbf{Q}$ represents the exchange's cost of holding inventories. The exchange provides more liquidity when the eigenvalues of this matrix are smaller. The eigenvalues of the matrix are the diagonal elements when $\mathbf{Q}$ is a diagonal matrix.

The modified maximization problem is

$$\max_{\mathbf{x},\mathbf{y}} \left[\mathbf{x}^\mathsf{T}\mathbf{p}^H - \tfrac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{D}\,\mathbf{x} + \mathbf{y}^\mathsf{T}\bar{\boldsymbol{\pi}} - \tfrac{1}{2}\mathbf{y}^\mathsf{T}\mathbf{Q}\,\mathbf{y}\right] \qquad \text{subject to} \qquad \mathbf{W}\mathbf{x}+\mathbf{y} = \mathbf{0} \qquad \text{and} \qquad \mathbf{0} \le \mathbf{x} \le \mathbf{q}. \tag{28}$$

This problem differs from maximization problem (11) by adding the exchange's utility $\mathbf{y}^\mathsf{T}\bar{\boldsymbol{\pi}} - \frac{1}{2}\mathbf{y}^\mathsf{T}\mathbf{Q}\,\mathbf{y}$ to the objective and by adding the exchange's quantity traded $\mathbf{y}$ to the market clearing condition. If clearing prices are $\boldsymbol{\pi}^*$, the exchange acquires inventories $\mathbf{Q}^{-1}(\bar{\boldsymbol{\pi}} - \boldsymbol{\pi}^*)$. Note that the exchange's preferences are defined over assets while the customer orders define preferences over portfolios.[9]

Theorem 6. *The problem* (28) *has a unique solution for both optimal (market-clearing) quantities* $\mathbf{x}^*$, $\mathbf{y}^*$ *and for prices* $\boldsymbol{\pi}^*$.

Unlike our previous results in Theorems 1 and 2, Theorem 6 guarantees that the prices are unique by guaranteeing that there is always a partially executable order from the exchange, and these partially executable orders can be used to price every asset. In this way, trading by the exchange solves the tie-breaking problem.

Proof. The problem (28) can easily be converted into almost exactly the same form as the problem (11) when the exchange is treated as a customer, the exchange' orders for assets are modified to be orders for portfolios, and the vectors $\mathbf{x}$ and $\mathbf{y}$ are then stacked together. The only difference is that the lower limit price on the exchange's orders is close to minus-infinity and the exchange's execution rates are very large numbers (even though the actual execution rate is small because the orders are only barely partially executable relative to the lower limit price near infinity). Thus, Theorems 1 and 2 imply existence of unique market clearing quantities and possibly non-unique clearing prices. Since the exchange's orders are all partially executable and

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

the matrix **Q** is positive definite, the unique quantities traded by the exchange can be uniquely inverted to obtain the prices at which the exchange trades the optimal prices. □

Computation Algorithms. In addition to making prices unique, trading by the exchange also tends to make it easier for the algorithm to find clearing prices. Intuition for this is obtained by thinking of the gradient method of optimization as a multivariate generalization of Walrasian tatonnement. Algorithmically, Walrasian tatonnement corresponds to solving the dual problem by using excess demands at guessed price vectors to suggest the direction price changes for the next guess.

If the search direction for improving a price-vector guess is chosen to be proportional to uncleared quantities, as suggested by Walrasian tatonnement intuition, the resulting algorithm is the gradient method. Without exchange trading, the chosen direction may not reduce all uncleared quantities because the guessed prices may may not have any partially executable orders for some portfolios. If the exchange trades to maximize quadratic utility with invertible Q matrix, this guarantees that the exchange will be providing liquidity for all portfolios, and therefore small-enough changes in quantities will generate small but strict reductions in uncleared quantities for all portfolios.

Since the improvement from each guess may be very tiny, this intuition does not lead to a practical algorithm for calculating prices unless the exchange trades very aggressively. Mathematically, several issues make the gradient method impractical. First, different stock prices and levels of liquidity for different assets creates a scaling issue. A small price change may be needed to clear the market for a liquid asset with large excess demand while a large price change may be needed to clear the market for an illiquid asset with small excess demand. Second, portfolio orders potentially makes the hessian matrix poorly conditioned, creating issues similar to multicollinearity in a linear regression. Price changes for reaching market clearing prices need to take into account whether assets are complements or substitutes. Third, as prices change, orders can switch from being partally executable to being unexecutable or fully executable—or vice versa. This makes the hessian matrix change as constraints on different orders begin to bind or cease to bind. In addition to dealing with current excess demand, an effective algorithm also needs to take account of how order executability changes as price change.

These issues are all standard ones in the theory of convex optimization. Boyd (2004) discusses how interior point methods can be used to solve equality- and inequality-constrained, convex problems like ours. Interior point methods solve the KKT conditions simultaneously. They explicitly deal with the third issue above by implicitly using information on how binding contraints are to guide choice of search direction. They also have the following advantages (Boyd, 2004):

1. A bound can be placed on the number of iterations needed to reach a desired accuracy.

2. In theory and in practice, the number of iterations to achieve a given accuracy is predictable.

3. Solutions with high accuracy are obtained.

All of these characteristics are highly desirable for calculating flow-trading equilibrium because more predictability of computation time will result in fewer trading halts, and highly accurate prices (six or more significant digits) are needed if millions of dollars of trading takes place each second.

To examine numerical feasibility of flow trading, we simulate a book of orders for assets and portfolios, then use an optimization algorithm to calculated clearing prices. Our current simulation uses the Cvxopt package in a Python environment. Both Python and Cvxopt are open-source. Cvxopt solves quadratic programming problems using cone methods, a variation of interior point methods. This method solves the KKT equations by replacing the complementary slackness equations (19) with the modified conditions

$$\boldsymbol{\lambda}^* \cdot (\mathbf{q} - \mathbf{x}^*) = \bar{\mu}, \qquad \boldsymbol{\mu}^* \cdot \mathbf{x}^* = \bar{\mu}, \qquad \bar{\mu} > 0, \qquad \bar{\mu} \to 0. \tag{29}$$

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

In each iteration of the Cvxopt algorithm, a different linearized system of equations must be solved for updates to $\mathbf{x}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\lambda}$. This requires solving a different positive-definite linear system with $3I + N$ unknowns on each iteration. Solving this system usually dominates computation time in the algorithm. The algorithm gives the user a choice between using a default solver or defining a user-supplied custom solver. Our problem has two features which make a custom solver particularly desirable. First, the number of orders I is likely to be much larger than the number of assets N. Second, the matrix $\mathbf{D}$, which the algorithm assumes to be a positive-definite symmetric matrix, is in our case diagonal and therefore easily inverted. This allows us to use a custom solver which uses the explicit inverse of $\mathbf{D}$ to substitute out updates to the variables $\mathbf{x}$, $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$, reducing the number of variables in the linear system from $3I + N$ to only N. This dramatically speeds up the algorithm.

For example, our base-case simulation assumes $N = 100$ assets and $I = 15000$ orders. Our user-supplied solver calculates a Cholesky decomposition of a 100×100 matrix, which is far faster than the Cholesky decomposition of the 45100×45100 matrix using the default algorithm.

In the Cvxopt algorithm, most of the operations involving assets are matrix-vector products, which can be parallelized. Important calculations involving prices are matrix decompositions (Cholesky), which are less easily parallelized. Therefore, we expect that increases in computation time related to increases in the number of orders will not be a big problem because more cores can be used to perform computations with more orders. Increasing computation time related to increasing the number of assets may prove to be more challenging because adding more computing resources may not speed up the $O(N^3)$ operations required for matrix decompositions as easily.

Intuition suggests using the solution from the previous second's auction to "warm start" a search for the solution for the next second's auction. One disadvantage of the Cvxopt algorithm is that the parameter $\bar{\mu}$ must also be reset when the optimization is restarted. The affects the appropriate starting values for multipliers and makes it difficult to warm-start the algorithm.

4.3 Simulations

We have implemented a simulation framework to test the practical performance of flow trading. Calculations are performed on a Lenovo P1 laptop with 6 cores (12 threads) operated at 2.0 mh. The goal of the simulations is to show that a base-case problem with 100 assets and 15000 orders can reliably be solved in one second. We regard meeting this goal as "proof of concept" that flow trading is computationally feasible.

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

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

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

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

troduces a potential difficulty for an optimizer. The problem is more difficult if there are many such portfolio orders.

3. Changing hessian: To make the hessian change dramatically as prices change, the average difference $\mathbf{p}^H - \mathbf{p}^L$ can be made small. When $\mathbf{p}^H - \mathbf{p}^L$ is large, market depth does not change much as prices changes, the problem looks linear to an optimizer, so clearing prices are likely to be easy for the optimizer to find. When $\mathbf{p}^H - \mathbf{p}^L$ is small, the range over which an order is partially executable small, and the hessian changes greatly when prices change a small amount. In the limit $\mathbf{p}^H - \mathbf{p}^L \to 0$, the hessian itself goes to zero for a randomly guessed price vector.

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

The problem difficulty can also be varied by varying the intensity of exchange trading. The problem is easier when the exchange provides a deeper market for all assets.

The order book is constructed using parameters listed in Table 2, which provides a name for the parameter, a base-case assumption for the parameter value, and a description of the parameter. The number of assets is 100. There are 5000 orders for individual assets, 5000 orders for various indexes, and 5000 orders which randomly buy an index or asset and sell an equal expected dollar value of another asset or index asset. In addition to value-weighted- and equally-weighted market indexes, there are also 5 indexes for dollar volume quantiles and 10 indexes for industries. Assets are assigned to volume and industry indexes by sorting the assets by expected trading volume, then assigning the assets to value quantiles by their rank in the sort and assigning assets to "industry" quantiles according to their rank modulo the number of industry indexes. The dollar-volume shares of the equally-weighted index orders, volume-weighted quantile index orders, equally-weighted quantile index orders, volume-weighted industry-index orders, and equally-weighted industry-index orders are 0.05, 0.08, 0.02, 0.08, and 0.02, respectively; the remaining 0.75 share of index volume is volume-weighted index orders.

Positive, continuous random variables are assumed to have a log-normal distribution. All index prices are normalized to have an expected price of $100.00 and log-variance of 0.50. The expected price is defined as the mean of the order midpoint $(p_i^H + p_i^L)/2$. There is no concept of fundamental value; prices depend only on orders. Whether the share price is $1.00 or $100.00 should not affect equilibrium prices, but the number and dollar size of orders matters a great deal.

We use the market microstructure invariance assumptions of Kyle and Obizhaeva (2016) to convert assumptions about the dollar volume all orders if fully executable into assumptions about the number and dollar size of orders. For any asset or index, the dollar volume of orders

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

Variable	Value	Description
N	100	number of assets
MA	5000	number of orders for individual assets
MX	5000	number of orders for indexes
$M2$	5000	number of pairs-trade orders to swap assets or indexes
$num_quantile_indexes$	5	number industry quantiles
$num_industry_indexes$	10	number of industries
ew_index_share	0.05	dollar volume share of equally-weighted index orders
$vw_quantile_index_share$	0.08	dollar volume share of value-weighted index orders
$ew_quantile_index_share$	0.02	dollar volume share of equally-weighted index orders
$ew_industry_index_share$	0.08	dollar volume share of equally-weighted index orders
$ew_industry_index_share$	0.02	dollar volume share of equally-weighted index orders
$index_price$	\$100.00	price of one share of any index
$std_num_orders_asset$	1.7	log-standard deviation of dollar order size for given asset
$mean_asset_price$	\$100.00	mean expected price of each asset
std_asset_price	0.50	log-standard deviation of asset price
$invariance_exponent$	1/3	invariance exponent
$invariance_c$	\$2000.00	invariance constant
$invariance_m$	0.60	invariance moment ratio
std_order_size	1.5	log-standard deviation of dollar order size for given asset
std_limit_price	0.10	log standard deviation of order's limit price midpoint
$fraction_buy_orders_asset$	0.50	fraction of asset orders which buy an asset
$fraction_buy_orders_index$	100	fraction of index orders which buy the index
$avg_ph_minus_pl_bp$	1.00	$(2 * p_i^H - p_i^L)/(p_i^H + p_i^L) * 10^{-4}$, $p_i^H - p_i^L$ in basis points
$std_ph_minus_pl$	2.50	log standard deviation of $p_i^H - p_i^L$
$frac_exchange_liquidity$	100	larger value implies exchange provides more liquidity

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

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

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

The average difference between p_i^H and p_i^L is assumed to be 1.00 basis point. For a typical order to trade and index, this assumption implies that we expect p_i^H = \$99.995 and p_i^L = \$100.005. This calibration is designed to stress the algorithm, not to capture realistic expectations about market operation.

Finally, there is a difficult-to-interpret parameter, which defines how aggressively the exchange trades each asset. A larger value of this parameter results in the exchange market-making orders capturing a larger fraction of volume. The baseline value of 0.00001 leads to little trading by the exchange.

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

In addition, the optimizer runs on only one core at the low speed of 2.0 gh.

An important implementation detail concerns the manner in which portfolios are aggregated. If there are hundreds of orders to trade a particular portfolio, such as the value-weighted index, instead of multiplying by the same portfolio weights hundreds of times, the algorithm first adds up the portfolio demands for the particular portfolio across orders, then multiplies by the portfolio weights only one time. This improves computational efficiency dramatically because multiplying portfolio weights by orders executions rates are the biggest computational bottleneck when the number of assets is less than a few hundred. In futures implementations,

Table 3: Base-case results of simulations of flow trading. The table presents statistics for number of iterations ($iterations$), number of seconds to calculate clearing prices (dt), number of basis points of uncleared dollar volume absorbed by the exchange ($exchBP$), and number of basis points of volume traded by the exchange with market-making orders ($exchBP$). For example, $exchBP$ = 0.02 means that the exchange trades $0.02 for every $10000 traded, which is equivalent to $2 for every $1 million in dollar volume. The variables with suffix $_polished$ are values of the same outcomes after up to five additional iterations which polish the solution using a Cvxopt option providing more accurate but slower matrix algebra.

	mean	max	min	std	count
iterations	27.3700	30.0000	22.0000	2.0807	1000
dt	0.6188	0.8547	0.4857	0.0515	1000
unclrdBP	0.5874	37.8490	0.0000	2.1110	1000
exchBP	0.0275	0.0712	0.0077	0.0096	1000
iterations_polished	4.9010	5.0000	2.0000	0.3942	1000
dt_polished	0.1890	0.2639	0.0834	0.0164	1000
unclrdBP_polished	0.0112	4.7032	0.0000	0.1632	1000
exchBP_polished	0.0275	0.0712	0.0077	0.0096	1000

we plan to parallelize these computations across multiple cores.

Results. Summary results of the base-case simulations of 1000 order books are presented in Table 3. Processing time varies between 0.3857 seconds and 0.8547 seconds. With the maximum number of iterations set to 30, the actual number of iterations has a mean of 27.37; the maximum of 30 iterations is frequently binding. The exchange does almost no trading with market making orders. Uncleared volume absorbed by the exchange is a maximum of 0.37849 percent of volume. With up to five further iterations to polish the solution, which take on average 0.1890 seconds, the worst case for uncleared volume falls to 0.047032 percent of volume. It could be reduced even closer to zero with more iterations. We interpret these results as being a proof of concept in the sense that clearing prices for a market with 100 assets and 15000 orders can be calculated in one second. In a production environment, faster CPU speeds, parallelized matrix operations, and more refined algorithms should make it easy to calculate clearing prices in much less than one second.

The simulation framework makes it easy to change assumptions about parameters. Preliminary results suggest that computation time is potentially dominated by two bottlenecks. One bottleneck scales linearly with the number of orders. We believe that this bottleneck can be ameliorated by parallel processing of these calculations. Another bottleneck, related to matrix decompositions of $N \times N$ matrices, scales with the cube of the number of assets. The second bottleneck is not important with 100 assets but does become important with several hundred

assets. It may be harder to overcome with parallel processing.

Preliminary results also suggest that a tiny fraction of exchange trading speeds up calculations a great deal by reducing the number of iterations. Finding clearing prices also takes fewer iterations when the difference between p_i^H and p_i^L is larger.

5 Microfoundations for Portfolio Orders

We require that flow orders specify downward-sloping demand curves for an arbitrary portfolio. The demand—the rate of trade—depends only on the portfolio's price. Despite this restriction, flow orders can replicate the optimal portfolio when traders have CARA preferences. Flow orders can still approximate the optimal portfolio locally for more general preferences, but orders must be updated if prices change significantly.

The restriction that demand curves for portfolios must be downward sloping is natural in many applications.

Our framework assumes quasilinear utility functions of the form $u(\mathbf{x}) - \boldsymbol{\pi}^\mathsf{T}\mathbf{x}$. This function has asset demands $\mathbf{x}$ but, without loss of generality, omits wealth because there are no wealth effects. As a function of wealth $\mathbf{W}$, the exponential utility function $-\mathrm{e}^{-A\mathbf{W}}$, frequently used in finance, does not have wealth effects either. To fit exponential utility into our framework, consider the problem of maximizing transformed exponential utility

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} \left[-\frac{1}{A} \log\left(\mathbb{E}\left[-\exp^{-A(\mathbf{v}-\boldsymbol{\pi})\mathbf{x}}\right]\right) \right]. \tag{30}$$

Since the cost of the portfolio $\boldsymbol{\pi}^\mathsf{T}\mathbf{x}$ is nonstochastic, it factors out, so that this problem has the form of our quasilinear utility function

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} \left[u(\mathbf{x}) - \boldsymbol{\pi}^\mathsf{T}\mathbf{x} \right], \qquad \text{where} \qquad u(\mathbf{x}) := -\frac{1}{A} \log\left(\mathbb{E}_\mathbf{v}\left[\exp^{-A\mathbf{v}\mathbf{x}}\right]\right). \tag{31}$$

The function $u(\mathbf{x})$ is concave in $\mathbf{x}$.[10] For jointly normally distributed random vectors $\mathbf{v}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the utility function redeces to a standard mean-variance portfolio optimization problem

$$u(\mathbf{x}) - \boldsymbol{\pi}^\mathsf{T}\mathbf{x} = \boldsymbol{\mu}^\mathsf{T}\mathbf{x} - \tfrac{1}{2} A \mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{x} - \boldsymbol{\pi}^\mathsf{T}\mathbf{x}. \tag{32}$$

Here $\boldsymbol{\mu}$ is a vector of subjective expected "fundamental values" (dollars per share), $\mathbf{x}$ is a

[10] Concavity can be proved using Jensen's inequality to switch the order of the log function and expectation operator.

vector of asset weights (shares), $\mathbf{\Sigma}$ is a covariance matrix of returns, A measures risk aversion (per dollar), and $\boldsymbol{\pi}$ is a vector of market prices (dollars per share). The cost of the portfolio is $\boldsymbol{\pi}^{\mathsf{T}}\mathbf{x}$ (dollars).

The optimal portfolio is

$$\mathbf{x}^* = A^{-1}\mathbf{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{\pi}). \tag{33}$$

This solution is a linear demand curve, in which the quantity demanded for one asset depends on the prices of all other assets through the inverse of the positive (semi) definite matrix $\mathbf{\Sigma}$.[11] This demand curve has no wealth effects.

The demand curves do not match our requirement that the demand for a portfolio depends only on the price of the portfolio, not on the price of all assets in some arbitrary manner. To deal with this issue, we can use a singular value decomposition of the covariance matrix $\mathbf{\Sigma}$ to rotate the assets so that the demand for each rotated asset is a linear function of the price of the rotated asset itself and not the price of other assets. This rotated demand will be (weakly) downward sloping since the covariance matrix is negative (semi) definite.

The singular value decomposition allows $\mathbf{\Sigma}$ to be written as

$$\mathbf{\Sigma} = \mathbf{WDW}^{\mathsf{T}}, \tag{34}$$

where $\mathbf{W}$ is an orthogonal matrix and $\mathbf{D}$ is a diagonal matrix of singular values. We will now show that the matrix $\mathbf{W}$ is exactly the matrix of relevant portfolio weights, consistent with our previous notation $\mathbf{W}$. For an arbitrary matrix, the singular value decomposition is $\mathbf{UDW}^{\mathsf{T}}$, where $\mathbf{U}$ and $\mathbf{W}$ are different orthogonal matrices and $\mathbf{D}$ is an arbitrary diagonal matrix. The fact that the covariance matrix $\mathbf{\Sigma}$ is positive (semi) definite makes $\mathbf{U} = \mathbf{W}$ and makes all of the diagonal elements of $\mathbf{D}$ positive (nonnegative). We can think of $\mathbf{W}$ as a matrix of portfolios which rotates the assets so that they become orthogonal factors. The inverse of $\mathbf{\Sigma}$ is given by $\mathbf{\Sigma}^{-1} = \mathbf{WD}^{-1}\mathbf{W}^{\mathsf{T}}$ (due to the fact that the orthonormal matrix $\mathbf{W}$ is its own inverse). The matrix $\mathbf{D}^{-1}$ is a diagonal matrix whose entries are reciprocals of the diagonal entries of $\mathbf{D}$. Due to the orthogonality properties of the singular value decomposition, $\mathbf{\Sigma}$ can be expressed as the sum of rotated factor

[11] When the matrix $\mathbf{\Sigma}$ is positive semidefinite but not positive definite, we can use the pseudo-inverse instead of the inverse to define the demand function. The use of the pseudo-inverse reduces the dimensionality of the demand function.

covariance matrices of rank one each:[12]

$$\mathbf{\Sigma} = \sum_{n=1}^{N} D_n \mathbf{w}_n {\mathbf{w}_n}^{\mathsf{T}}, \tag{35}$$

where $\mathbf{w}_n$ is the nth row of $\mathbf{W}$ and D_n is the ith diagonal entry of $\mathbf{D}$. Thus, demand can be written

$$\mathbf{w}^* = A^{-1} \sum_{n=1}^{N} D_n^{-1} \mathbf{w}_n {\mathbf{w}_n}^{\mathsf{T}} (\boldsymbol{\mu} - \boldsymbol{\pi}). \tag{36}$$

In the above expression, $\mathbf{w}_n {\mathbf{w}_n}^{\mathsf{T}}$ is a positive semidefinite matrix. Using associativity of matrix multiplication, this can be written

$$\mathbf{w}^* = A^{-1} \sum_{n=1}^{N} D_n^{-1} \cdot \mathbf{w}_n \cdot ({\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\mu} - {\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\pi}), \tag{37}$$

where ${\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\mu} - {\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\pi}$ is a scalar.

Now the above portfolio $\mathbf{w}^*$ can be implemented with n portfolio flow orders. The nth portfolio order results in a vector of asset quantities given by

$$q_n \cdot \text{trunc}\left(\frac{{\mathbf{p}^H}_n - {\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\pi}}{{\mathbf{p}^H}_n - {\mathbf{p}^L}_n} \right) \cdot \mathbf{w}_n. \tag{38}$$

Choose a very high upper limit price and very low lower limit price ${\mathbf{p}^L}_n << {\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\pi} << {\mathbf{p}^H}_n$ so that the order is fully executable. Since the trunc() function is not binding, the order purchases the quantity

$$\frac{q_n}{{\mathbf{p}^H}_n - {\mathbf{p}^L}_n} \cdot ({\mathbf{p}^H}_n - {\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\pi}) \cdot \mathbf{w}_n \tag{39}$$

over the relevant range of prices. Now choose the order quantity q_n so that $\frac{q_n}{{\mathbf{p}^H}_n - {\mathbf{p}^L}_n} = A^{-1} D_n^{-1}$. This set of N flow orders implements the optimal portfolio. Furthermore, the matrix $\mathbf{W}$ is exactly the matrix of relevant portfolio weights, consistent with our previous notation $\mathbf{W}$.

The key insight here is the following: In the expression $A^{-1} D_n^{-1} \cdot \mathbf{w}_n \cdot ({\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\mu} - {\mathbf{w}_n}^{\mathsf{T}} \boldsymbol{\pi})$, the fact that the covariance matrix $\mathbf{\Sigma}$ is positive definite makes the first occurrence of $\mathbf{w}_n$, which measures the assets bought by the portfolio, exactly the same as the second occurrence of $\mathbf{w}_n$, which measures how the quantity demanded depends on the price of a relevant portfolio, which flow trading requires to be the same. This allows the optimal portfolio to be implemented with N flow trading orders, each of which is a downward-sloping demand curve in which the quantity

[12] If $\mathbf{\Sigma}$ is positive semidefinite but not positive definite, at least one singular value is zero. In this case, the inverse is not defined, but the pseudoinverse is defined. The pseudoinverse changes the nonzero elements to $\mathbf{D}$ to their reciprocals but leaves the zero elements unchanged. In numerical applications, where numbers theoretically equal to zero may have small nonzero values due to rounding error, the very small singular values are changed to zero.

of the portfolio depends only on the price of the portfolio. The absence of income effects and the risk aversion assumption that less variance is better than more allow the solution to fit our framework, which only requires that the matrix $\mathbf{\Sigma}$ be positive (semi) definite.

It may seem an unlikely coincidence that the matrix $\mathbf{W}$ is orthonormal. The explanation is that our notation for demand curves has redundant parameters that allow us to scale the problem so that the matrix $\mathbf{W}$ is orthonormal. The scaling appears in the diagonal matrix $\mathbf{D}$.

Approximation for General Preferences The logic of this CARA-normal example can be applied more generally. For an arbitrary strictly concave twice continuously differentiable quasi-linear utility function over assets, the optimal demand can be locally approximated by the inverse of the second derivative of the utility function. Since the utility function is strictly concave, the local linear approximation is a negative-semidefinite linear operator, corresponding to an $N \times N$ matrix in the spaces of asset prices and quantities. Thus, we can use the singular value decomposition argument from above to obtain the locally approximated demand as a set of N flow orders for portfolios of N assets. As market prices change, the local approximation changes, in which case the trader will have to revise the portfolio orders. A similar approach can still be used when there is asymmetric information or when there is price impact.

To summarize, negative semidefinite matrices have two important properties. First, from the definition of negative semidefinite, demand curves slope weakly downward no matter how assets are rotated. Second, if assets are rotated according to a singular value decomposition, the demand for a portfolio is a downward sloping function of the portfolio price and does not depend on the prices of other orthogonally rotated portfolios.

Limitations. Our approach of requiring orders to be downward sloping demand curves of linear combinations of assets is restrictive. Whether the restriction is reasonable is a practical problem, depending on the trading strategies traders want to pursue. Generalizing order types is an issue for further research.

We do not allow an order to buy one asset as a function of the price of another asset that is not itself traded as part of the order. Such an order is not consistent with a concave utility function because it implies nonconcave preferences. The Hessian matrix may not be negative definite.

We do not allow orders with flat segments in the inverse demand curve—the derivative of price with respect to quantity is zero. Similarly, we do not allow portfolio orders that treat two different assets as perfect substitutes and therefore force the assets to trade at the same price over the range of portfolio quantities where the order is partially executable. When the assets are perfect substitutes for all traders, this is addressed by combining the perfect substitutes

into a single asset. When this is not possible, the trader may place an order to buy an equally weighted portfolio of the two assets and simultaneously place two pairs-trade orders which buy the cheaper of the two assets and sells the more expensive one. This achieves a similar result but is somewhat less convenient because the trader must manage the three portfolio orders.

We conjecture that not every concave utility function can be approximated by orders for portfolios. Proving this conjecture in a mathematically precise manner is current work in progress.

6 Discussion of Implementation and Policy Issues

6.1 Implementation: Auction Frequency and Outcome Discovery

The frequency of market clearing is an important choice in the market design. Traders with more complex dynamic strategies may prefer a short batch interval because it allows greater information with which to better optimize dynamic strategies. The good news is that there is little cost to the short batch interval. A trader with a simple strategy can submit a single persistent order and ignore all the dynamics. There is no harm from the extra information and control that the shorter batch interval allows.

The cost of a shorter batch interval is that it allows less time for computation. In situation where it is desirable for the exchange to include more computationally difficult order types then a longer batch interval likely is needed.

We envision a variety of batch intervals. The equities trade in fragmented exchanges likely requires a short batch interval, say every second, but other markets may work fine with batch intervals of one minute, one hour, or one day. A benefit of flow trading is the batch interval is easily adjusted to circumstances. For example, an exchange can shift from trading every second to trading every even second without any adjustments to orders. Thus, if congestion made computation too difficult, the exchange could automatically move to even second intervals. Auction managers in multi-round auctions, such as spectrum auctions, routinely adjust the frequency of rounds to manage the auction pace and improve price and outcome discovery.

The trade-offs of auction frequency are illustrated in wholesale electricity markets where a version of flow trading is used in practice (Cramton, 2017). In the real-time energy market, generators submit persistent piecewise-linear supply curves that are the basis for market-clearing every five minutes. Generators can adjust their supply curves throughout the day in response to changing circumstances. Thus, the order book evolves but every 5-minutes the market is cleared to determine the flows over the next five minutes and the real-time energy price. Auction intervals of 5-minutes are used to accommodate the significant computation required. More importantly, there is little advantage from higher frequencies since the relevant products

are simultaneously optimized in each batch auction. Further, the system operator procures a special service, frequency regulation, that accommodates deviations at the one-second frequency.

Wholesale electricity markets also illustrate how flow trading can be used to improve forward energy markets (Cramton, 2021). Forward energy trading plays a critical role in letting participants manage risk. Real-time electricity prices are highly volatile, ranging from -250to9000 per megawatt-hour. Demand-side participants reduce risk by purchasing in advance a quantity that is consistent with expected consumption. Likewise, suppliers are happy to sell most of their anticipated production in forward markets. Flow trading would allow market participants to trade thousands of forward products with ease. For example, a service provider has obligations to serve its customers. The provider estimates its load shape and desires to enter the real-time market with a position consistent with its expected demand. With flow trading, the service provider could place a single order for a portfolio of thousands of products to move from its current position to a new target. A linear combination of monthly forward energy contracts by hour (00 to 23) and day type (weekday and weekend) could be adjusted to approximate any load shape. Within the month, hourly forward energy contracts could further refine matching either demand obligations or production capabilities in advance of real time. Flow trading enables continuous matching of demand to supply as circumstances change due to weather, outages, or other factors—both in real time as occurs today and days, months, and years ahead.

6.2 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, we propose no pre-trade transparency.[13]

If auctions are held at fixed times, such as the beginning of the first millisecond of every second, then pretrade liquidity information will be subject to change if traders are allowed to cancel new orders within the second prior to an auction. This can lead to gaming incentives to post and cancel orders as cheap talk. To avoid such gaming incentives, there are two potential solutions: Either avoid publishing pretrade information about orders, or hold auctions at random times, with say a one-second expected time interval between each auction. Holding auctions at random times eliminates cheap talk incentives by making all orders actionable since they may be executed any time. We do not favor publishing information about orders which may be canceled before execution at no cost.

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. Providing extensive information about a large number of portfolios is both impractical—because it requires enormous bandwidth—and inefficient—because it encourages an arms race in which faster and more sophisticated traders gain at the expense of slower and less sophisticated traders. Instead of publishing as much information as traders would demand, an 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.

Without detailed information about the shape of the order book, liquidity might be different at different price points due to random locations of upper and lower limit prices on orders. If the exchange publishes a great deal of information about the shape of the order book, a likely profitable strategy is to place orders at price levels where liquidity is low and avoid placing orders at price points where liquidity is high. Without such information, traders will have incentives to

[13] An alternative mechanism for implementing frequent batch auctions is to hold auctions at random times generated by a Poisson process with an arrival rate of once per second. Using the method, it would theoretically make sense for the exchange to publish tentative market prices because they are actionable in that trade might occur immediately. Nevertheless, the gains would be negligible and the computational costs would likely make this impossible to implement in real-time.

implement a substitute strategy of placing orders where there is a large difference between the upper limit price $\mathbf{p}^H$ and the lower limit price $\mathbf{p}^L$. This has the beneficial side effect of making the order book more "linear" and therefore make finding clearing prices easier.

Markets implement rules for both pre-trade and post-trade transparency. At a minimum, a flow trading design publishes one clearing price and one market-clearing quantity for each asset traded at each batch auction. For example, with 8000 ticker symbols and trading hours from 9:30 a.m. to 4:00 p.m. (23 400 seconds per day), there are 187.2 million "prints" per day. It is also possible for the exchange to publish data about the depth of the order book both pre-trade and post trade. For example, price impact costs can be precisely measured relative to what prices would have been over the execution interval if the order had not been placed and other traders placed the same orders they actually placed.

It is reasonable for the exchange itself, following a published rule, to submit orders to provide limited liquidity by trading on the exchange's own account. This can have two advantages. First, if the exchange places a small linear order in each asset—for example with flow quantity a linear function of price and zero trade at the most recent price—clearing prices become unique because the implied utility function becomes globally strictly convex. Second, the computation time for calculating equilibrium quantities and prices is reduced. The exchange's orders also define a bound on the speed at which an optimizer converges to a clearing price vector.

If market clearing quantities and prices are not completed within one second, the exchange could declare the optimizer's best estimate of market prices as the official price, with the exchange itself trading net uncleared quantities to force market clearing. In practical situations, such quantities are likely to be small because an algorithm's estimate of clearing prices will be close to actual clearing prices when the optimization process terminates before reaching the optimum.

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.

Flow trading also improves trust by reducing front-running. Flow orders execute gradually.

A would-be front-runner must trade along with the order, not ahead of it.

With flow trading, the exchange can confirm proper receipt and execution of an order by reporting quantity traded. By contrast, with current markets, not only must each trader receive a confirmation that its order was executed but also each non-executed order that would have been executable at the market price requires a confirmation that it was not executed. Auctions with discrete price and quantity grids, such that quantity allocations are not unique, are not only unfair—in the sense of punishing slower traders who cannot afford an arms race—but also informationally wasteful.

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.

With flow trading, there still are trading costs. Trading faster requires offering better prices, which makes clearing pricesmove, which creates price impact.

Fragmentation and Competition. Current markets are fragmented in two ways. First, multiple exchanges post quotes and compete for order flow. Second, a significant fraction of orders are executed off-exchange. Currently, many brokers receive payment for order flow in exchange for routing retail orders to off-exchange execution venues. The result is that orders are not matched with one another, even if they are willing to trade at the same time at the same prices. Instead, customers incur a bid-ask spread cost.

An exchange with flow orders has a competitive advantage over an exchange with traditional orders in two ways. First, traders provide liquidity to one another by trading at the same price.

Second, flow traders can pick off the traditional orders based on information about executing flow orders, but traders cannot easily pick off flow orders based on information about current displayed traditional orders.

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 or 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

Flow trading allows participants to express preferences in a powerful and flexible way.

We anticipate that flow trading can reduce trading costs and improve gains from trade in financial securities. The result is a more efficient market for financial securities and lower trading costs for fundamental investors.

References

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. 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.

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

Cramton, Peter. 2021. "Forward Energy Market." University of Cologne Working Paper.

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.

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

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." *The 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*.

Mas-Colell, Andreu, Michael Dennis Whinston, and Jerry R Green. 1995. *Microeconomic theory*. Vol. 1, Oxford university press New York.

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

Parkes, David C., Christopher Thorpe, and Wei Li. 2015. "Achieving Trust without Disclosure: Dark Pools and a Role for Secrecy-Preserving Verification." *Proceedings of the Third Conference on Auctions,MarketMechanisms and Their Applications*.

Parkes, David C., Michael O. Rabin, and Christopher Thorpe. 2009. "Cryptographic Combinatorial Clock-Proxy Auctions." *Financial Cryptography and Data Security*, 305–324.

Parkes, David C., Michael O. Rabin, Stuart M. Shieber, and Christopher A. Thorpe. 2008. "Practical Secrecy-preserving, Verifiably Correct and Trustworthy Auctions." *Electronic Commerce Research and Applications*, 7(3): 294–312.

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

Rostek, Marzena J, and Ji Hee Yoon. 2020*b*. "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.

Thorpe, Christopher, and David C. Parkes. 2007. "Cryptographic Securities Exchanges." *Proc. 11th International Conference on Financial Cryptography and Data Security*, 163–178.

Thorpe, Christopher, and David C. Parkes. 2009. "Cryptographic Combinatorial Securities Exchanges." *Financial Cryptography and Data Security*, 285–304.

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.

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