The Design of Financial Exchanges: Some Open Questions at the Intersection of Econ and CS

Eric Budish, University of Chicago

Simons Institute Conference: Algorithmic Game Theory and Practice

Nov 2015

Overview

1. The economic case for discrete-time trading
   - Financial exchange design that is predominant around the world – continuous limit order book – is economically flawed
   - Flaw: treats time as a continuous variable (serial processing)
   - Solution: treat time as a discrete variable, batch process using an auction. "Frequent batch auctions".
   - Eric Budish, Peter Cramton and John Shim ("BCS") 2015, *Quarterly Journal of Economics*
2. The computational case for discrete-time trading
   - Discrete time respects the limits of computers and communications technology. Not infinitely fast.
   - Benefits for exchanges, algo traders, regulators
   - Qualitative/informal argument in BCS 2015, would benefit greatly from Econ/CS research
3. Other Econ/CS Questions about the Design of Financial Exchanges
   - Flash crashes
   - Speed vs. Smarts Tradeoff
   - Circuit Breakers

Overview

1. **The economic case for discrete-time trading**
   - **Financial exchange design that is predominant around the world – continuous limit order book – is economically flawed**
   - **Flaw: treats time as a continuous variable (serial processing)**
   - **Solution: treat time as a discrete variable, batch process using an auction. "Frequent batch auctions".**
   - **Eric Budish, Peter Cramton and John Shim ("BCS") 2015, *Quarterly Journal of Economics***
2. The computational case for discrete-time trading
   - Discrete time respects the limits of computers and communications technology. Not infinitely fast.
   - Benefits for exchanges, algo traders, regulators
   - Qualitative/informal argument in BCS 2015, would benefit greatly from Econ/CS research
3. Other Econ/CS Questions about the Design of Financial Exchanges
   - Flash crashes
   - Speed vs. Smarts Tradeoff

Brief Description of the Continuous Limit Order Book

- Basic building block: limit order
  - Specifies a price, quantity, and buy/sell (bid/ask)
  - "Buy 100 shares of XYZ at \$100.00"
- Traders may submit limit orders to the market at any time during the trading day
  - Also may cancel or modify outstanding limit orders at any time
  - Orders and cancelations are processed by the exchange one-at-a-time in order of receipt (serial process)
- Set of outstanding orders is known as the limit order book
- Trade occurs whenever a new limit order is submitted that is either (i) bid $\geq$ lowest ask; (ii) ask $\leq$ highest bid
  - New limit order is interpreted as accepting (fully or partially) one or more outstanding orders

BCS Model: Basics of Setup

- There is a security, x, that trades on a continuous limit-order book market
- There is a publicly observable signal, y, of the value of security x. y evolves as a Poisson jump process.
- Purposefully strong assumption:
  - Fundamental value of x is *perfectly* correlated to the public signal y
  - x can always be costlessly liquidated at this fundamental value
- Players
  - Investors: arrive stochastically, mechanically either buy or sell x at market
  - Trading Firms: N, all equally fast – zero latency
- Overall: "Best case" scenario for price discovery and liquidity provision
  - No asymmetric info, no inventory costs, investors mechanical, trading firms have zero latency

"Sniping"

- Given the model setup – no asymmetric information, no inventory costs, etc. – one might conjecture that (Bertrand) competition among the N trading firms leads to effectively infinite liquidity for investors
- That is, trading firms should offer to buy or sell x at price y in unlimited quantity at zero bid-ask spread
- But that is not what happens in the continuous limit order book market, due to a phenomenon we call "sniping"

“Sniping”

- Suppose y jumps, e.g., from $\underline{y}$ to $\bar{y}$
- Trading firms providing liquidity in the market for x send a message to the continuous limit order book
  - Withdraw old quotes based on $\underline{y}$
  - Replace with new quotes based on $\bar{y}$

"Sniping"

- However, at the exact same time, *other* trading firms send a message to the continuous market attempting to "snipe" the stale quotes before they are adjusted
  - Buy at the old quotes based on $\underline{y}$, before these quotes are withdrawn
- Since the continuous market processes messages in *serial* – that is, one at a time – it is possible that a message to snipe a stale quote will get processed before the message to adjust the stale quote
- In fact, not only possible but *probable*
  - For every 1 liquidity provider trying to get out of the way
  - $N-1$ other trading firms trying to snipe him
  - Hence, when there is a big jump, liquidity provider gets sniped with probability $\frac{N-1}{N}$

BCS Model: 3 Key Takeaways about Continuous Markets

BCS Model: 3 Key Takeaways about Continuous Markets

1. Mechanical arbitrage opportunities are "built in" to the market design
    - *Symmetrically observed public information creates arbitrage rents.*
    - This isn't supposed to happen in an efficient market. (Fama, 1970)
    - OK to make money from asymmetric information, but symmetric information is supposed to get into prices for free. Market failure.

BCS Model: 3 Key Takeaways about Continuous Markets

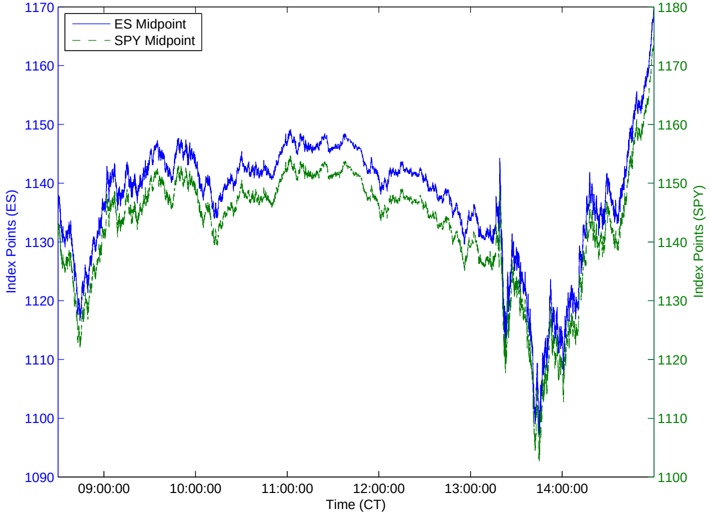
1. Mechanical arbitrage opportunities are "built in" to the market design
   - *Symmetrically observed public information creates arbitrage rents.*
   - This isn't supposed to happen in an efficient market. (Fama, 1970)
   - OK to make money from asymmetric information, but symmetric information is supposed to get into prices for free. Market failure.
2. Profits from mechanical arbs come at the expense of liquidity provision
   - In a competitive market, sniping costs get passed on to investors.
   - Thinner markets, wider bid-ask spreads.

BCS Model: 3 Key Takeaways about Continuous Markets

1. Mechanical arbitrage opportunities are "built in" to the market design
   - *Symmetrically observed public information creates arbitrage rents.*
   - This isn't supposed to happen in an efficient market. (Fama, 1970)
   - OK to make money from asymmetric information, but symmetric information is supposed to get into prices for free. Market failure.
2. Profits from mechanical arbs come at the expense of liquidity provision
   - In a competitive market, sniping costs get passed on to investors.
   - Thinner markets, wider bid-ask spreads.
3. Sniping creates a never-ending race for speed
   - Sniping: win race to pick off stale quotes.
   - Liquidity provision: get out of the way of the snipers!

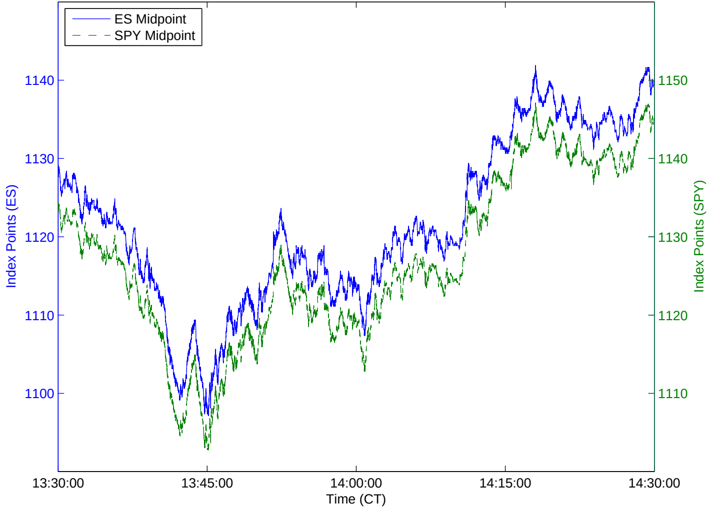
Mechanical Arbitrage Example: S&P 500 Index Arb

ES vs. SPY: 1 Day



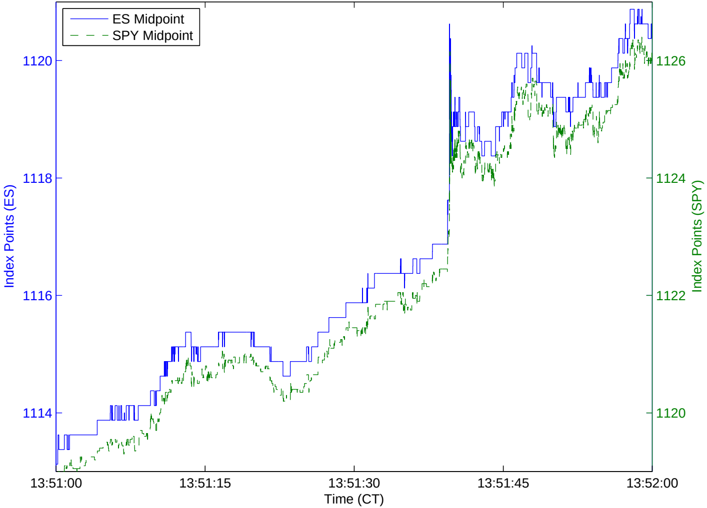
Mechanical Arbitrage Example: S&P 500 Index Arb

ES vs. SPY: 1 hour



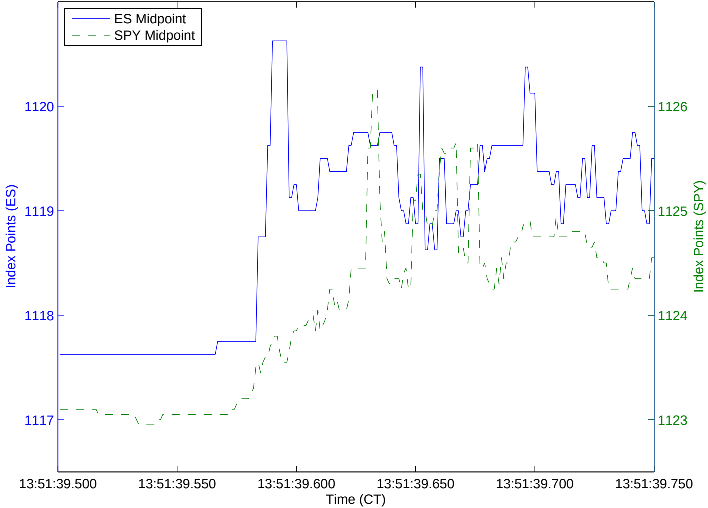
Mechanical Arbitrage Example: S&P 500 Index Arb

ES vs. SPY: 1 minute



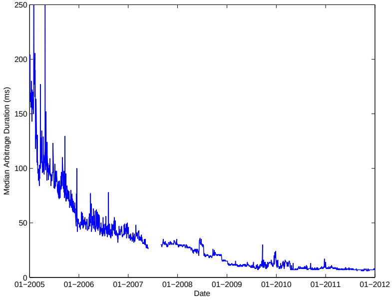
Mechanical Arbitrage Example: S&P 500 Index Arb

ES vs. SPY: 250 milliseconds

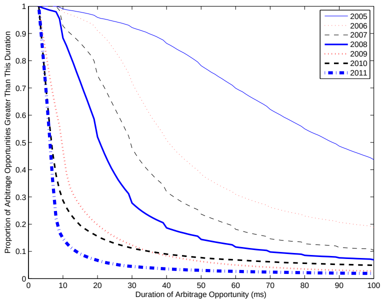


Arb Durations over Time: 2005-2011

Median over time

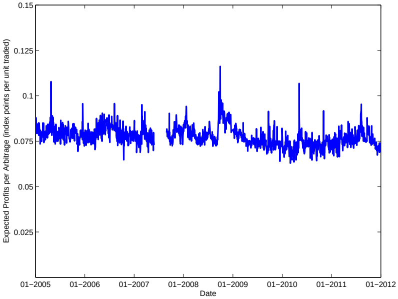


Distribution by year

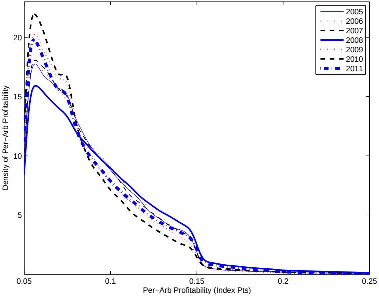


Arb Per-Unit Profits over Time: 2005-2011

Median over time

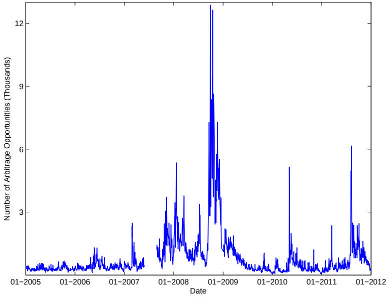


Distribution by year

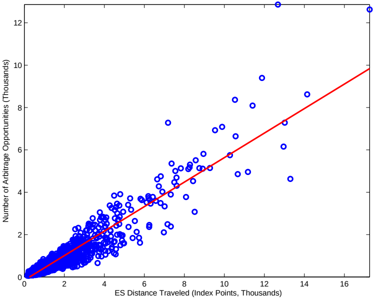


Arb Frequency over Time: 2005-2011

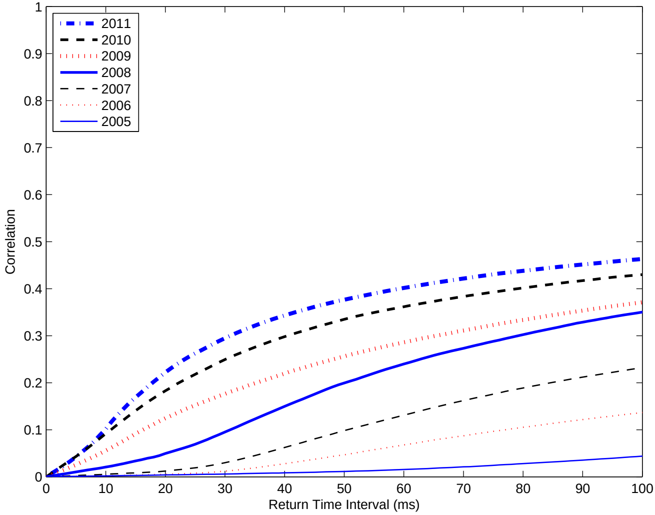
Frequency over time



Frequency vs. Volatility



Correlation Breakdown Over Time: 2005-2011



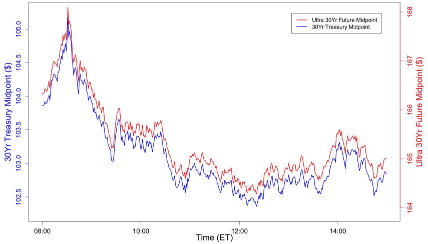
Latency Arb and Arms Race are "Constants" of the Market Design

To summarize:

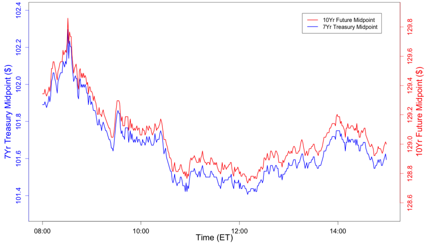
- Competition <u>does</u> increase the speed requirements for capturing arbs ("raises the bar")
- Competition <u>does not</u> reduce the size or frequency of arb opportunities
- Suggests we should think of latency arbitrage and the resulting arms race as a "constant" of the current market design

Analogy to the US Treasury Market

30 year ultra future vs. 30 year cash



10 year future vs. 7 year cash



Other Highly Correlated Pairs

Partial List

E-mini S&P 500 Futures (ES) vs. SPDR S&P 500 ETF (SPY)
E-mini S&P 500 Futures (ES) vs. iShares S&P 500 ETF (IVV)
E-mini S&P 500 Futures (ES) vs. Vanguard S&P 500 ETF (VOO)
E-mini S&P 500 Futures (ES) vs. ProShares Ultra (2x) S&P 500 ETF (SSO)
E-mini S&P 500 Futures (ES) vs. ProShares UltraPro (3x) S&P 500 ETF (UPRO)
E-mini S&P 500 Futures (ES) vs. ProShares Short S&P 500 ETF (SH)
E-mini S&P 500 Futures (ES) vs. ProShares Ultra (2x) Short S&P 500 ETF (SDS)
E-mini S&P 500 Futures (ES) vs. ProShares UltraPro (3x) Short S&P 500 ETF (SPXU)
E-mini S&P 500 Futures (ES) vs. 500 Constituent Stocks
E-mini S&P 500 Futures (ES) vs. 9 Select Sector SPDR ETFs
E-mini S&P 500 Futures (ES) vs. E-mini Dow Futures (YM)
E-mini S&P 500 Futures (ES) vs. E-mini Nasdaq 100 Futures (NQ)
E-mini S&P 500 Futures (ES) vs. E-mini S&P MidCap 400 Futures (EMD)
E-mini S&P 500 Futures (ES) vs. Russell 2000 Index Mini Futures (TF)
E-mini Dow Futures (YM) vs. SPDR Dow Jones Industrial Average ETF (DIA)
E-mini Dow Futures (YM) vs. ProShares Ultra (2x) Dow 30 ETF (DDM)
E-mini Dow Futures (YM) vs. ProShares UltraPro (3x) Dow 30 ETF (UDOW)
E-mini Dow Futures (YM) vs. ProShares Short Dow 30 ETF (DOG)
E-mini Dow Futures (YM) vs. ProShares Ultra (2x) Short Dow 30 ETF (DXD)
E-mini Dow Futures (YM) vs. ProShares UltraPro (3x) Short Dow 30 ETF (SDOW)
E-mini Dow Futures (YM) vs. 30 Constituent Stocks
E-mini Nasdaq 100 Futures (NQ) vs. ProShares QQQ Trust ETF (QQQ)
E-mini Nasdaq 100 Futures (NQ) vs. Technology Select Sector SPDR (XLK)
E-mini Nasdaq 100 Futures (NQ) vs. 100 Constituent Stocks
Russell 2000 Index Mini Futures (TF) vs. iShares Russell 2000 ETF (IWM)
Euro Stoxx 50 Futures (FESX) vs. Xetra DAX Futures (FDAX)
Euro Stoxx 50 Futures (FESX) vs. CAC 40 Futures (FCE)
Euro Stoxx 50 Futures (FESX) vs. iShares MSCI EAFE Index Fund (EFA)
Nikkei 225 Futures (NIY) vs. MSCI Japan Index Fund (EWJ)
Financial Sector SPDR (XLF) vs. Constituents
Financial Sector SPDR (XLF) vs. Direxion Daily Financial Bull 3x (FAS)
Energy Sector SPDR (XLE) vs. Constituents
Industrial Sector SPDR (XLI) vs. Constituents
Cons. Staples Sector SPDR (XLP) vs. Constituents
Materials Sector SPDR (XLB) vs. Constituents
Utilities Sector SPDR (XLU) vs. Constituents
Technology Sector SPDR (XLK) vs. Constituents
Health Care Sector SPDR (XLV) vs. Constituents
Cons. Discretionary Sector SPDR (XLY) vs. Constituents
SPDR Homebuilders ETF (XHB) vs. Constituents
SPDR S&P 500 Retail ETF (XRT) vs. Constituents
Euro FX Futures (6E) vs. Spot EURUSD
Japanese Yen Futures (6J) vs. Spot USDJPY
British Pound Futures (6B) vs. Spot GBPUSD
Australian Dollar Futures (6B) vs. Spot AUDUSD
Swiss Franc Futures (6S) vs. Spot USDCHF
Canadian Dollar Futures (6C) vs. Spot USDCAD
Gold Futures (GC) vs. miNY Gold Futures (QO)
Gold Futures (GC) vs. Spot Gold (XAUUSD)
Gold Futures (GC) vs. E-micro Gold Futures (MGC)
Gold Futures (GC) vs. SPDR Gold Trust (GLD)
Gold Futures (GC) vs. iShares Gold Trust (IAU)
miNY Gold Futures (QO) vs. E-micro Gold Futures (MGC)
miNY Gold Futures (QO) vs. Spot Gold (XAUUSD)
miNY Gold Futures (QO) vs. SPDR Gold Trust (GLD)
miNY Gold Futures (QO) vs. iShares Gold Trust (IAU)
E-micro Gold Futures (MGC) vs. SPDR Gold Trust (GLD)
E-micro Gold Futures (MGC) vs. iShares Gold Trust (IAU)
E-micro Gold Futures (MGC) vs. Spot Gold (XAUUSD)
Market Vectors Gold Miners (GDX) vs. Direxion Daily Gold Miners Bull 3x (NUGT)
Silver Futures (SI) vs. miNY Silver Futures (QI)
Silver Futures (SI) vs. iShares Silver Trust (SLV)
Silver Futures (SI) vs. Spot Silver (XAGUSD)
miNY Silver Futures (QI) vs. iShares Silver Trust (SLV)
miNY Silver Futures (QI) vs. Spot Silver (XAGUSD)
Platinum Futures (PL) vs. Spot Platinum (XPTUSD)
Palladium Futures (PA) vs. Spot Palladium (XPDUSD)
Eurodollar Futures Front Month (ED) vs. (12 back month contracts)
10 Yr Treasury Note Futures (ZN) vs. 5 Yr Treasury Note Futures (ZF)
10 Yr Treasury Note Futures (ZN) vs. 30 Yr Treasury Bond Futures (ZB)
10 Yr Treasury Note Futures (ZN) vs. 7-10 Yr Treasury Note
2 Yr Treasury Note Futures (ZT) vs. 1-2 Yr Treasury Note
2 Yr Treasury Note Futures (ZT) vs. iShares Barclays 1-3 Yr Treasury Fund (SHY)
5 Yr Treasury Note Futures (ZF) vs. 4-5 Yr Treasury Note
30 Yr Treasury Bond Futures (ZB) vs. iShares Barclays 20 Yr Treasury Fund (TLT)
30 Yr Treasury Bond Futures (ZB) vs. ProShares UltraShort 20 Yr Treasury Fund (TBT)
30 Yr Treasury Bond Futures (ZB) vs. ProShares Short 20 Year Treasury Fund (TBF)
30 Yr Treasury Bond Futures (ZB) vs. 15+ Yr Treasury Bond
Crude Oil Futures Front Month (CL) vs. (6 back month contracts)
Crude Oil Futures (CL) vs. ICE Brent Crude (B)
Crude Oil Futures (CL) vs. United States Oil Fund (USO)
Crude Oil Futures (CL) vs. ProShares Ultra DJ-UBS Crude Oil (UCO)
Crude Oil Futures (CL) vs. iPath S&P Crude Oil Index (OIL)
ICE Brent Crude Front Month (B) vs. (6 back month contracts)
ICE Brent Crude (B) vs. United States Oil Fund (USO)
ICE Brent Crude (B) vs. ProShares Ultra DJ-UBS Crude Oil (UCO)
ICE Brent Crude (B) vs. iPath S&P Crude Oil Index (OIL)
Natural Gas (Henry Hub) Futures (NG) vs. United States Nat Gas Fund (UNG)

Frequent Batch Auctions: Overview

- High level: identical to the current market design but for two key differences
  - Time is treated as discrete, not continuous
  - Orders are processed in batch, using an auction, not serially in order of arrival

Frequent Batch Auctions: Definition

- During the batch interval (eg 100ms) traders submits bids and asks
  - Can be freely modified, withdrawn, etc.
  - If an order is not executed in the batch at time t, it automatically carries over for $t+1, t+2, \ldots,$
  - Just like standard limit orders

Frequent Batch Auctions: Definition

- During the batch interval (eg 100ms) traders submits bids and asks
  - Can be freely modified, withdrawn, etc.
  - If an order is not executed in the batch at time t, it automatically carries over for $t+1, t+2, \ldots,$
  - Just like standard limit orders
- At the end of each interval, the exchange "batches" all of the outstanding orders, and computes market-level supply and demand curves

Frequent Batch Auctions: Definition

- During the batch interval (eg 100ms) traders submits bids and asks
  - Can be freely modified, withdrawn, etc.
  - If an order is not executed in the batch at time t, it automatically carries over for $t+1, t+2, \ldots,$
  - Just like standard limit orders
- At the end of each interval, the exchange "batches" all of the outstanding orders, and computes market-level supply and demand curves
- If supply and demand intersect, then all transactions occur at the same market-clearing price ("uniform price")

Frequent Batch Auctions: Definition

- During the batch interval (eg 100ms) traders submits bids and asks
  - Can be freely modified, withdrawn, etc.
  - If an order is not executed in the batch at time t, it automatically carries over for $t+1, t+2, \ldots,$
  - Just like standard limit orders
- At the end of each interval, the exchange "batches" all of the outstanding orders, and computes market-level supply and demand curves
- If supply and demand intersect, then all transactions occur at the same market-clearing price ("uniform price")
- Priority: still price-time, but treat time as discrete. Orders submitted in the same batch interval have the same priority. Pro-rata to break ties.

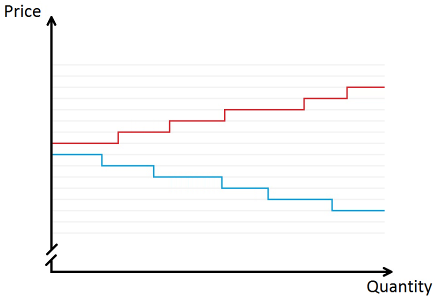
Frequent Batch Auctions: Definition

- During the batch interval (eg 100ms) traders submits bids and asks
  - Can be freely modified, withdrawn, etc.
  - If an order is not executed in the batch at time t, it automatically carries over for $t+1, t+2, \ldots,$
  - Just like standard limit orders
- At the end of each interval, the exchange "batches" all of the outstanding orders, and computes market-level supply and demand curves
- If supply and demand intersect, then all transactions occur at the same market-clearing price ("uniform price")
- Priority: still price-time, but treat time as discrete. Orders submitted in the same batch interval have the same priority. Pro-rata to break ties.
- Information policy: info is disseminated in discrete time. After each auction, all orders active for the auction displayed publicly
  - Activity during the interval is not displayed publicly (gaming)
  - Discrete time analogue of current practice in a CLOB market

Frequent Batch Auctions: 3 Cases

Case 1: Nothing happens during the batch interval

- Very common case: most instruments, most 100ms periods (or shorter), there is zero trade
- All outstanding orders carry forward to next interval
- Analogous to displayed liquidity in a LOB market

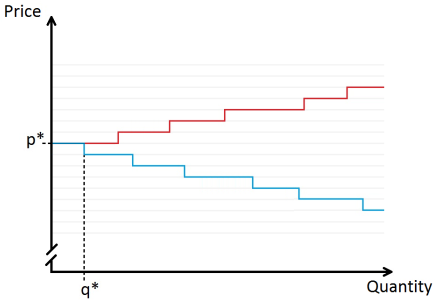


Case 1: No Trade

Frequent Batch Auctions: 3 Cases

Case 2: Small amount of trade

- Example: an investor arrives wanting to buy a small amount at market
- Demand will cross supply at the bottom of the supply curve
- Analogous to trading at the ask in a LOB market



Case 2: Investor Buys q*

Frequent Batch Auctions: 3 Cases

Case 3: Burst of activity in the interval

- Example: there is public news and many algos respond
- In this case, continuous and discrete are importantly different

Frequent Batch Auctions: 3 Cases

Case 3: Burst of activity in the interval

- Example: there is public news and many algos respond
- In this case, continuous and discrete are importantly different
- Continuous: process burst of activity based on order of receipt: competition on speed

Frequent Batch Auctions: 3 Cases

Case 3: Burst of activity in the interval

- Example: there is public news and many algos respond
- In this case, continuous and discrete are importantly different
- Continuous: process burst of activity based on order of receipt: competition on speed
- Discrete: process burst of activity using an auction: competition on price

Frequent Batch Auctions: 3 Cases

Case 3: Burst of activity in the interval

- Example: there is public news and many algos respond
- In this case, continuous and discrete are importantly different
- Continuous: process burst of activity based on order of receipt: competition on speed
- Discrete: process burst of activity using an auction: competition on price
- Helps liquidity in 2 ways

Frequent Batch Auctions: 3 Cases

Case 3: Burst of activity in the interval

- Example: there is public news and many algos respond
- In this case, continuous and discrete are importantly different
- Continuous: process burst of activity based on order of receipt: competition on speed
- Discrete: process burst of activity using an auction: competition on price
- Helps liquidity in 2 ways
  1. Liquidity providers have until end of interval to adjust their quotes to reflect new info
     - Being tiny bit slower than competition almost never matters

Frequent Batch Auctions: 3 Cases

Case 3: Burst of activity in the interval

- Example: there is public news and many algos respond
- In this case, continuous and discrete are importantly different
- Continuous: process burst of activity based on order of receipt: competition on speed
- Discrete: process burst of activity using an auction: competition on price
- Helps liquidity in 2 ways
  1. Liquidity providers have until end of interval to adjust their quotes to reflect new info
     - Being tiny bit slower than competition almost never matters
  2. Liquidity providers are protected by the auction: get a market consensus price based on new info
     - No more sniping. Public information induces price competition, not speed competition

Equilibrium Costs and Benefits of Frequent Batch Auctions

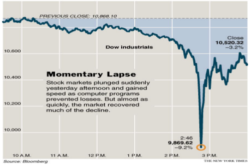
- Benefits
  - Enhanced liquidity
  - Eliminate socially wasteful arms race
- Costs
  - Investors must wait until the end of the batch interval to transact

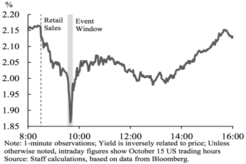
[see paper for formal equilibrium statements]

Overview

1. The economic case for discrete-time trading
   - Financial exchange design that is predominant around the world – continuous limit order book – is economically flawed
   - Flaw: treats time as a continuous variable (serial processing)
   - Solution: treat time as a discrete variable, batch process using an auction. "Frequent batch auctions".
   - Eric Budish, Peter Cramton and John Shim ("BCS") 2015, *Quarterly Journal of Economics*
2. **The computational case for discrete-time trading**
   - **Discrete time respects the limits of computers and communications technology. Not infinitely fast.**
   - **Benefits for exchanges, algo traders, regulators**
   - **Qualitative/informal argument in BCS 2015, would benefit greatly from Econ/CS research**
3. Other Econ/CS Questions about the Design of Financial Exchanges
   - Flash crashes
   - Speed vs. Smarts Tradeoff
   - Circuit Breakers

Context: Computational Issues in Modern Financal Markets



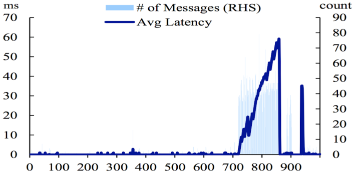


- Flash Crashes
  - 5/6/2010 equity market
  - 10/15/2014 US treasury market
  - numerous “mini” flash crashes in individual stocks
- Exchange glitches / outages
  - Nasdaq outage Aug 2013
  - Facebook IPO glitch May 2012
  - CME backlog during treasury flash crash
- Knight capital coding error (-$440M in 45mins) -> brink of bankruptcy
- Policy controversies with computational component: Data feeds (Flash Boys), Clock Synchronization

Computational Benefits of Discrete Time

- Overall claim/conjecture: discrete-time trading has significant computational simplicity benefits for markets relative to continuous-time trading.
- I'll describe specific benefits for:
  1. Exchanges
  2. Algorithmic Traders
  3. Regulators / Market observers
- Main intuition
  - Continuous-time markets implicitly assume that computers and communications technology are infinitely fast.
  - Computers and communications are fast but not infinitely so.
  - Discrete time respects the limits of computers and communications.
- Caveat: argument is qualitative / informal. Would benefit from more formal treatment. (Whole point of this talk is to encourage such work).

Computational Benefits of Discrete Time: Exchanges



Note: 1-millisecond intervals; Include new, modify, delete; All US Treasury futures contracts
Source: Staff calculations, based on data from CME Group.

- Continuous time: Exchange backlog
  - Processing any single order is computationally trivial
  - Disseminating any single message is computationally trivial
  - Even trivial operations take strictly positive time
  - Implication: during surges of activity, backlog and processing delay (Ex: 10/15/2014)
  - Algorithms left uncertain about state of the market and their own orders, precisely during times of heavy activity
- Discrete time
  - Batch auctions are also computationally trivial ($O(n \log n)$)
  - Can set batch interval to be long relative to realistic worst case processing time
- Research question: economic consequences of backlog

Computational Benefits of Discrete Time: Exchanges

High-Speed Traders Exploit Loophole

High-speed traders are using a hidden facet of the Chicago Mercantile Exchange's computer system to trade on the direction of the futures market before other investors get the same information.

- May 1,2013 WSJ

- Continuous time: message processing details economically important
  - CME: sends "trade updates" before "book updates".
  - Leads to practice of sentinel orders to privately learn about big price moves before the market as a whole
- Discrete time: can disseminate all messages at economically the same time.

Computational Benefits of Discrete Time: Exchanges

Nasdaq in Fresh Market Failure

Glitch That Locked Out Investors for Three Hours Is Latest Malfunction in Electronic Trading

- Aug 22,2013 WSJ

- Continuous time: numerous exchange glitches
  - Facebook IPO
  - Nasdaq Outage
  - Numerous smaller incidents
- Discrete time: exchange programming is simpler in many ways
  - Another example: NASDAQ threads all activity across all symbols through a single core to preserve exact sequence of events. Unnecessary once time discrete
- Research question: is there a connection between computational simplicity and these extreme events / glitches?

Computational Benefits of Discrete Time: Algorithms

"There are rules you need to follow to write fast code. Don't touch the kernel. Don't touch main memory . . . Don't branch."

-HFT Interview in MacKenzie (2014)

- Basic contrast between continuous time and discrete time
- Continuous time: get one piece of data at a time, respond
  - Fundamental tradeoff: speed versus smarts
  - How comprehensively does the algo "think" about the information before responding?
- Discrete time: get a batch of data from time t, make decisions for time $t+1$
  - Still have tradeoff speed vs. smarts for thinking that takes longer than one batch interval
  - But "the first 100 milliseconds are free!" No cost in terms of time priority.

Computational Benefits of Discrete Time: Algorithms

"It took him 52 seconds to realise what was happening, something was terribly wrong, and he pressed the red button ... By then we had lost \$3 million. ... in another twenty seconds ... the trading firm would have been bankrupt, and in another fifty or so seconds, our clearing broker would have been bankrupt ... "

-HFT Interview in MacKenzie (2014)

- Research questions

1. How to model speed vs. smarts: seems hard.
2. Are there negative externalities from prioritizing speed > smarts?
   - Dumb/fast decisions will cost the algo money, is there a broader harm?
3. How is the accuracy of prices affected by the speed vs. smarts tradeoff?
   - Prominent HFT's "strobe light" metaphor as argument for continuous > discrete

Computational Benefits of Discrete Time: Regulator

> *[T]he importance of data is further complicated by the many sources of data that must be aggregated . . . Varied data conventions, differing methods of communication, the sheer volume of quotes, orders, and trades produced each second, and even inherent time lags based on the laws of physics add yet more complexity.*
>
> *-SEC/CFTC Flash Crash Report*

- Paper Trail (aka audit trail)
- Continuous time: have to adjust the paper trail for
  - Relativity
  - Exchange latency
  - Exchange clock noise
- Non-trivial to figure out: did event A happen before, after or same time as event B?
- Discrete time: this becomes trivial

Computational Benefits of Discrete Time: Regulator

> *[T]he importance of data is further complicated by the many sources of data that must be aggregated . . . Varied data conventions, differing methods of communication, the sheer volume of quotes, orders, and trades produced each second, and even inherent time lags based on the laws of physics add yet more complexity.*
>
> *-SEC/CFTC Flash Crash Report*

- Research question: benefits of a clean paper trail
  - My own intuition: benefits could be large.
  - Analogy: value of a website simplifying its user interface can be large.
  - But: also case that this is just simplifying presentation of information that "sophisticated market participants already know".
  - Aside: it would be great to have better theory on why UI design is so important.

Computational Benefits of Discrete Time: Regulator

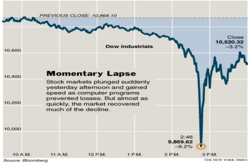
> *[W]e've been focusing on . . . situations that . . . give elite groups of traders access to market-moving information at the expense of the rest of the market. This is what we call Insider Trading 2.0, and it's one of the greatest threats to public confidence in the markets.*
>
> *-Eric Schneiderman, New York Attorney General*

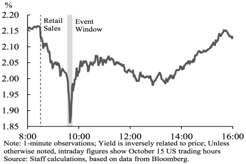
- Symmetric dissemination of public information
- Continuous time
  - Technologically infeasible to disseminate info such that all market participants who wish to receive it at the same time
  - Ex 1: Securities Information Processor (SIP) vs. Direct Feed controversy (Hendershott et al, 2013; Flash Boys)
  - Ex 2: SEC's dissemination of public filings via EDGAR website (Rogers, Skinner, Zechman)
  - Moreover: even if it was technologically possible to disseminate info to all at exactly the same time, theory model says that, economically, still asymmetric
- Discrete time: technologically trivial

Overview

1. The economic case for discrete-time trading
   - Financial exchange design that is predominant around the world – continuous limit order book – is economically flawed
   - Flaw: treats time as a continuous variable (serial processing)
   - Solution: treat time as a discrete variable, batch process using an auction. "Frequent batch auctions".
   - Eric Budish, Peter Cramton and John Shim ("BCS") 2015, *Quarterly Journal of Economics*
2. The computational case for discrete-time trading
   - Discrete time respects the limits of computers and communications technology. Not infinitely fast.
   - Benefits for exchanges, algo traders, regulators
   - Qualitative/informal argument in BCS 2015, would benefit greatly from Econ/CS research
3. **Other Econ/CS Questions about the Design of Financial Exchanges**
   - **Flash crashes**
   - **Speed vs. Smarts Tradeoff**
   - **Circuit Breakers**

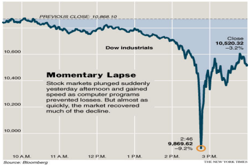
Flash Crashes

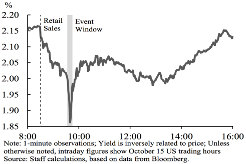




- Q1: what causes a flash crash?
  - Rational outcome given optimizing algos? Mistakes?
  - Relationship to speed vs. smarts discussion earlier?
  - Note: important to study both the major flash crashes (5/6/2010, 10/15/2014) and also the "mini" flash crashes that occur much more regularly in individual stocks

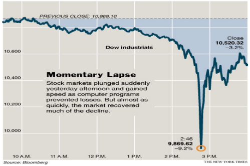
Flash Crashes

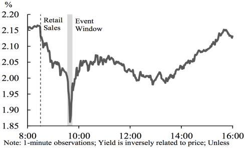




- Q2: what are the economic consequences of a flash crash?
  - does it matter if prices are wrong for a short period of time?
  - is there a potential for a dangerous "feedback loop" if flash crash occurs at end of trading day (close to 4pm) rather than in middle of day?
  - do extreme events like flash crashes affect investors' confidence / participation in the market? affect the cost of capital?

Flash Crashes





Note: 1-minute observations; Yield is inversely related to price; Unless otherwise noted, intraday figures show October 15 US trading hours
Source: Staff calculations, based on data from Bloomberg.

- Q3: are there exchange designs or other rules that could make exchanges less vulnerable to flash crashes?
- FBA specifically (speculative):
  - No sniping -> more liquidity -> less likely that a large order wipes out the order book
  - Discrete time -> more smarts vs. speed -> more likely that algos will not simply "turn off" in extreme events but rather will (smartly) take advantage of them

Circuit Breakers

- Idea: if there's an extreme market move take a "pause" before resuming trade
  - Example: SEC's Limit Up Limit Down mechanism
- Q1: is the basic rationale correct?
  - If it takes time to think in duress, why not allow time to think in quiet times too?
  - If argument is in quiet times there is no reason to restrict trade (mutually voluntary), fine, why restrict trades in duress?
- Q2: design details
  - Related assets. Ex: stocks and ETFs on 8/24/2015
  - In context of FBA: if auction clears far from last price, triggers circuit breaker, do those trades go through or not?

Conclusion

- Recap
  - Heart of BCS paper is economic case for frequent batch auctions
    - Eliminate rents from public information (sniping)
    - Enhances liquidity, stops socially wasteful arms race
  - BCS also make a computational case for frequent batch auctions
    - Arguments informal: no theorems, no data. Included because of importance of the topic
    - Lots of open research questions
    - Would naturally benefit from mix of Econ / CS tools

Conclusion

- Recap
  - Heart of BCS paper is economic case for frequent batch auctions
    - Eliminate rents from public information (sniping)
    - Enhances liquidity, stops socially wasteful arms race
  - BCS also make a computational case for frequent batch auctions
    - Arguments informal: no theorems, no data. Included because of importance of the topic
    - Lots of open research questions
    - Would naturally benefit from mix of Econ / CS tools
- Plea
  - More generally, there are lots of open questions about the design of financial exchanges.
  - Huge, important, markets.
  - Many questions will benefit from market design tools and approaches. Well worth paying the fixed cost to learn the relevant institutional details.