Algorithmic Trading with MATLAB®

Martin Demel, Application Engineer
Challenges when building trading strategies

- Increasing complexity
  - More data
  - More complicated models

- Increasing computational speed
  - Push to higher frequency

- Long deployment cycle
  - (Re)coding is costly and error-prone
Challenges through the organization
Customers using MATLAB

Asset Management
- BLACKROCK
- Fidelity Investments
- Monte dei Paschi di Siena
- Man Investments

Insurance
- Allianz
- Allstate
- Freddie Mac
- Swiss Re
- Manulife

Energy Trading
- bp
- UNION FENOSA
- EDISON
- Koch Industries Inc.
- e-on

Financial Services
- Deloitte
- Price Waterhouse Coopers
- Ernst & Young
- Standard & Poor’s
- Risk Metrics Group

Banks (Commercial, Retail, Investment)
- Citi
- Barclays Capital
- Bank of America
- UBS
- JPMorgan Chase & Co.
- State Street
- Lloyds Banking Group
- Goldman Sachs

Central Banks
- United States Federal Reserve
- International Monetary Fund
- Bank of Japan
Algorithmic trading workflow

Access
- Files
- Databases
- Datafeeds

Research and Quantify
- Data Analysis & Visualization
- Testing & Optimization
- Strategy Development

Automate

Share
- Reporting
- Applications
- Production

Files
- Databases
- Datafeeds

BZ | NG | CL
71.92 | 5.332 | 81.9

S=31; K=30
C=blsprice
F=C-S+K*ex

C/C++
Java
.NET
Algorithmic trading workflow

Access
- Files
- Databases
- Datafeeds

Research and Quantify
- Data Analysis and Visualization
- Testing and Optimization
- Strategy Development

Share
- Reporting
- Applications
- Production

Spreadsheet Link EX
- Database
- Datafeed

Fixed Income
- Financial Derivatives
- Econometrics

Financial
- Statistics
- Optimization

Builder EX
- Builder NE
- Builder JA

MATLAB Compiler

Report Generator

MATLAB
- Parallel Computing
- MATLAB Distributed Computing Server
Agenda

- Introduction: Algorithmic trading

- Developing an automated trading decision engine
  - Identify a successful trading rule
  - Extend trading rule set
  - Automate trading rule selection

- Break

- Implementing MATLAB into your production trading environment

- Wrap up and Q&A
The problem at hand: Identifying profitable trading strategies

- Commodities analyst
- Developing a trading strategy
  - Multiple trading rules
  - High frequency
- Management requirements:
  - Tested on historical data
  - Uses sophisticated analytics to identify optimal trading rule combination
  - Integrates with existing data and execution APIs
Trading decision engine

Development and testing

Historical Data
- End of Day / Intraday
- Files
- Databases

Strategy Modeling
- Research / Algorithms
- Model Development
- Calibration

Back Testing
- Profit / Loss
- Risk Exposure

Implementation

Live Data
- Real-Time Feeds
- Event-Based

Decision Engine
- Models
- Trading Rules

Execution
- Broker API
- Order Routing
Requirements for the trading engine

- Sophisticated analytics
  - Custom rules & indicators
  - Non-traditional techniques
- Scalable speed
  - Higher frequency data
  - More trading rules
- Quick to develop and deploy
  - Try different strategies
  - Embed in trading engine
Task 1: Develop a back testing environment

Goal: Build a back testing environment around historical data and a preliminary trading rule

Development and testing

- Historical Data
  - End of Day
  - Intraday
  - Files
  - Databases

- Strategy Modeling
  - Research / Algorithms
  - Model Development
  - Calibration

- Back Testing
  - Profit / Loss
  - Risk Exposure

Implementation

- Live Data
  - Real-Time Feeds
  - Event-Based

- Decision Engine
  - Models
  - Trading Rules

- Execution
  - Broker API
  - Order Routing
Key tasks

- Import data from files
- Create a preliminary rule
- Test the rule’s performance

Solutions

- MATLAB data tools
- High-level programming and pre-built functions
- Powerful graphics environment
Task 2: Expand the scale of the engine

Goal: Move to a higher frequency (minute-by-minute) and re-calibrate the model

Development and testing

Historical Data
- End of Day
- Intraday
- Files
- Databases

Strategy Modeling
- Research / Algorithms
- Model Development
- Calibration

Back Testing
- Profit / Loss
- Risk Exposure

Implementation

Live Data
- Real-Time Feeds
- Event-Based

Decision Engine
- Models
- Trading Rules

Execution
- Broker API
- Order Routing
Key tasks

- Importing data from databases
- Increase computational speed

Solutions

- MATLAB data tools: Database Toolbox
- High-performance computing: Parallel Computing Toolbox, MATLAB Distributed Computing Server
Task 3: Rule selection engine

Goal: Develop a rule selection system for instruments using evolutionary learning

Development and testing

- **Historical Data**
  - End of Day
  - Intraday
  - Files
  - Databases

- **Strategy Modeling**
  - Research / Algorithms
  - Model Development
  - Calibration

- **Decision Engine**
  - Models
  - Trading Rules

- **Back Testing**
  - Profit / Loss
  - Risk Exposure

Implementation

- **Live Data**
  - Real-Time Feeds
  - Event-Based

- **Execution**
  - Broker API
  - Order Routing
Key tasks

- Increase number of rules
- Incorporate advanced analytics to select best combination
Working with multiple strategies

Should I trade?

Signal 1
Yes

OR

Signal 2
Yes

OR

Signal 3
No

Working with multiple strategies

- Represent different combinations as *bit strings*
Building Custom Evolution Algorithms

- **Selection**
  - *Retain* the best performing bit strings from one generation to the next. *Favor these for reproduction*

- **Crossover**
  - parent1 = [1 0 1 0 0 1 1 0 0 0]
  - parent2 = [1 0 0 1 0 0 1 0 1 0]
  - child = [1 0 0 0 0 1 1 0 1 0]

- **Mutation**
  - parent = [1 0 1 0 0 1 1 0 0 0]
  - child = [0 1 0 1 0 1 0 0 0 1]
Key tasks

- Increase number of rules
- Incorporate advanced analytics to select best combination

Solutions

- High-level programming
- MATLAB Toolboxes: Global Optimization, …
Review: Requirements for the trading engine

- Sophisticated analytics
  - Custom rules & indicators
  - Non-traditional techniques
- Scalable speed
  - Higher frequency data
  - More trading rules
- Quick to develop and deploy
  - Try different strategies
  - Embed in trading engine
MATLAB’s solutions

- Sophisticated analytics
  - Advanced graphics environment
  - Toolboxes give access to hundreds of new techniques
  - Flexible and customizable

- Scalable speed
  - Parallel computing solution

- Quick to develop and deploy
  - High-level programming
  - Automated deployment… after the break
Agenda

- Introduction: Algorithmic trading
- Developing an automated trading decision engine
  - Identify a successful trading rule
  - Extend trading rule set
  - Automate trading rule selection

Break

- Implementing MATLAB into your production trading environment
- Wrap up and Q&A
Pairs Trading in Brief

- **Cointegration**: Two or more time series share *long-term* behavior
- Identify a pair that has spread apart
- Take opposing positions
- Profit occurs when prices revert
Key tasks / challenges

Key tasks
- Identify cointegrating relationships
- Test the strategy

Solution
- Econometrics Toolbox
  - New in R2011a: Engle-Granger and Johansen frameworks
- Code reuse from previous tasks
Implementing the Decision Engine

Goal: Evaluate and test the decision engine with real-time feeds and execution through a messaging bus

Development and testing

Historical Data
- End of Day / Intraday
- Files
- Databases

Strategy Modeling
- Research / Algorithms
- Model Development
- Calibration

Back Testing
- Profit / Loss
- Risk Exposure

Implementation

Live Data
- Real-Time Feeds
- Event-Based

Decision Engine
- Models
- Trading Rules

Execution
- Broker API
- Order Routing
Key Tasks

Key tasks
- Read live market data from data feed
- Connect to trading “engine”

Solutions
- Datafeed Toolbox
- Many external APIs
  - .NET, Java, C/C++, etc.
  - 3rd party APIs
Deploying Applications with MATLAB

- Give MATLAB code to other users

- Share applications with end users who do not need MATLAB
  - Stand-alone executables
  - Shared libraries
  - Software components
MATLAB’s solutions

- Sophisticated analytics
  - Advanced graphics environment
  - Toolboxes give access to hundreds of new techniques

- Scalable speed
  - Parallel computing solution

- Quick to develop and deploy
  - High-level programming
  - Automated deployment
Support and Community

MathWorks® | Book Program

MATLAB® Central

MathWorks® | Consulting Services

MathWorks® | Connections Program

MathWorks® | Training Services
Consulting Services
*Accelerating return on investment*

A global team of experts supporting every stage of tool and process integration
Training Services

*Exploit the full potential of MathWorks products*

Flexible delivery options:

- Public training available worldwide
- Onsite training with standard or customized courses
- Web-based training with live, interactive instructor-led courses

More than 30 course offerings:

- Introductory and intermediate training on MATLAB, Simulink, Stateflow, Real-Time Workshop, and PolySpace products
- Specialized courses in control design, signal processing, parallel computing, code generation, communications, financial analysis, and other areas
MATLAB Central

- Open exchange for the MATLAB and Simulink user community
- 662,000 visits per month
- File Exchange
  - Upload/download access to free files including MATLAB code, Simulink models, and documents
  - Ability to rate files, comment, and ask questions
  - More than 9,000 contributed files, 400 submissions per month, 25,500 downloads per day
- Newsgroup
  - Web forum for technical discussions about MATLAB and Simulink
  - 200 posts per day
- Blogs
  - Frequent posts from key MathWorks developers who design and build the products
  - Open conversation at blogs.mathworks.com

Based on February 2009 data
Connections Program

More than 300 add-on products and services that complement and extend MathWorks products:

- Specialized third-party toolboxes for MATLAB
- Interfaces to third-party software and hardware products
- Specialized training courses and consulting services
- System integrators and suppliers that incorporate MathWorks products
Book Program

More than 1,000 books for educational and professional use, in 26 languages

- Controls
- Signal Processing
- Image Processing
- Biosciences
- Communications
- Mechanical Engineering
- Mathematics
- Aerospace Engineering
- Environmental Sciences
- Chemistry
- Finance
- Electronics
Technical Support

Resources
- Over 100 support engineers
  - All with MS degrees (EE, ME, CS)
  - Local support in North America, Europe, and Asia
- Comprehensive, product-specific Web support resources

High customer satisfaction
- 95% of calls answered within three minutes
- 70% of issues resolved within 24 hours
- 80% of customers surveyed rate satisfaction at 80-100%