

The background features a dark blue field with several large, overlapping chevron shapes pointing to the right. The colors of these shapes include a bright orange, a medium blue, and a light cyan. The text is positioned on the left side of the image.

MathWorks
FINANCE
CONFERENCE 2023

October 11-12 | Online

MathWorks
FINANCE CONFERENCE 2023

Transformational Technologies: Empowering Financial Professionals with MATLAB

October 11-12 | Online

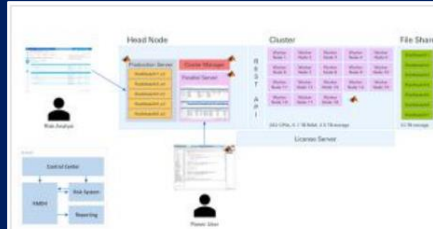


David Willingham,
Quantitative Finance Team Lead, MathWorks

From quote to sealing the deal, MATLAB is very well represented in the SP product chain



Kempen Takes Financial Documentation to the Next Level Using MATLAB



Swiss Re Migrates Risk Model to the Cloud with MATLAB Parallel Server

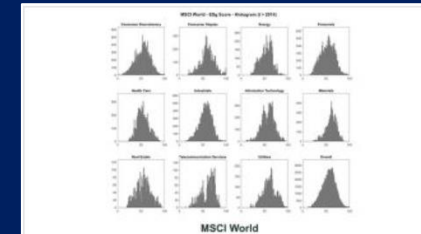


Southwest Airlines Simulates Fuel Market Movements in Hedging Strategy

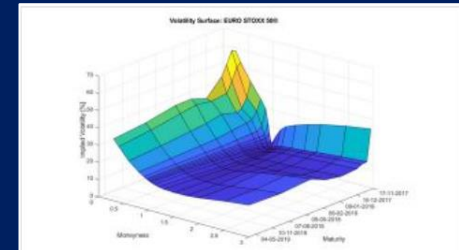
We do what we do to enable you to do what you do



FICC Department of China Galaxy Securities Valuates Financial Assets and Builds Trading and Hedging Strategies



State Street Global Advisors Develops Scoring Model to Bring Transparency to ESG Investing



Helaba Invest Develops and Deploys Enterprise Financial Analysis Software

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

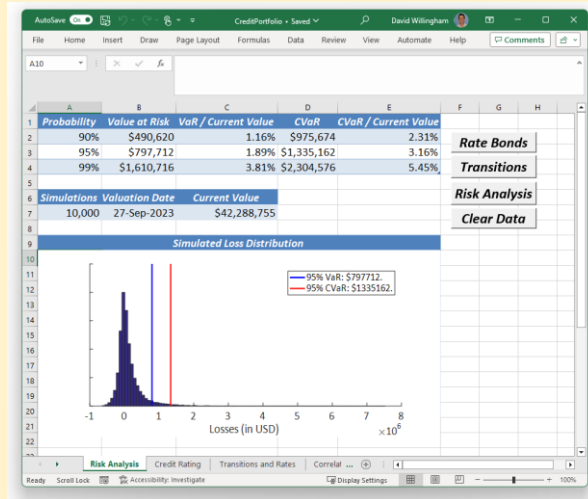
3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

Trend #1

Automation streamlines existing manual tasks and processes

Many tasks were done manually in excel



High level programming make it easier to automate

```
1 %% Backtest Performance of Maximum Information Ratio Investment Strategy
2 % In this example we evaluate the performance of the maximum portfolio ...
13
14
15 %% Import Data
16 % Import historical returns data for 44 large-cap stocks, the benchmark ...
20
21 load BlueChipStocks
22
23 iMarket = strcmpi('Market', Asset);
24 iCash = strcmpi('Cash', Asset);
25
26 Market = Data(:, iMarket);
27 Cash = Data(:, iCash);
28 Data = Data(:, ~iMarket & ~iCash);
29 Map = Map(:, ~iMarket & ~iCash);
30 Asset = Asset(:, ~iMarket & ~iCash);
31
32 clear iMarket iCash
33
34 %% Set up Backtest
35 % These parameters define the backtest framework
36
37 numportfolio = 20; % number of portfolios on each efficient frontier
38 window = 60; % historical estimation window in months
39 offset = 3; % shift in time for each frontier in months
40 cutoff = 0.4; % this fraction of data in a series must be non-NaN values
```

2005

2007

2009

2011

2013

2015

2017

2019

Today

How MATLAB programming has evolved from lines of code

```
%% Run backtest
% Loop through different intervals in time. For each interval,
tic;
for i = 1:nPeriods
    % set up date indices for current period
    startindex = endTimes(i) - window + 1;
    endindex = endTimes(i);
    % select "market" series
    Xmarket = Market(startindex:endindex);
    % select assets that are active on the endindex date
    iasset = Map(endindex,:);
    % keep series with sufficient numbers of non-NaN values
    imissing = sum(isnan(Data(startindex:endindex,:))) > cutoff*window;
    % form active universe for current endindex date
    iasset = logical(iasset) & ~logical(imissing);
    iasset(end-1:end) = 0; % last two series are not stocks (not used in this step)
    % select data for active universe
    A = Asset(iasset);
    X = Data(startindex:endindex,iasset);
    fprintf('Estimation period %s to %s with %d assets ...\n', ...
        datestr(Date(startindex)), datestr(Date(endindex)), numel(A));
    % remove "market" from the data (market-neutral relative returns)
    if relative
        X = X - repmat(Xmarket, 1, numel(A));
    end
    % construct portfolio object (use RiskFreeRate if not market-neutral)
    p = Portfolio('AssetList', A, 'Name', sprintf('Universe %s', datestr(Date(endindex))));
    if ~relative
        p = Portfolio(p, 'RiskFreeRate', Cash(endindex));
    end
    p = setDefaultConstraints(p);
    p = estimateAssetMoments(p, X, 'MissingData', true);
    % estimate portfolios on efficient frontier
    pwgt = estimateFrontier(p, numportfolio);
    % estimate portfolio that maximizes the ratio of relative risk to relative return
    % if absolute returns, then maximize the Sharpe ratio
    swgt = estimateMaxSharpeRatio(p);
    [srsk, sret] = estimatePortMoments(p, swgt);
    % display selected assets
    chosenAssets = p.AssetList(swgt > 1.0e-4);
    fprintf('\tSelected assets: %s', chosenAssets{1});
    fprintf(', %s', chosenAssets{2:end});
    fprintf('\n');
    % save data for 3D frontier
    PortDate(i) = Date(endindex);
    PortRisk(i,:) = sqrt(pfactor)*(estimatePortRisk(p, pwgt));
    PortReturn(i,:) = pfactor*(estimatePortReturn(p, pwgt));
    PortSigma(i) = sqrt(pfactor)*srsk;
    PortMean(i) = pfactor*sret;
    % evaluate performance
    if (endindex + offset) <= numel(Date)
        Xret = prod(1+Data (endindex+1:endindex+offset,:)) - 1;
        Marketret = prod(1+Market(endindex+1:endindex+offset)) - 1;
        Cashret = prod(1+Cash (endindex+1:endindex+offset)) - 1;
        PerfDate(i) = Date(endindex+offset);
        PerfData(i,:) = [Xret(iasset)*swgt Marketret Cashret];
    end
end
toc;
```

Backtesting ~2011

To using high level functions for common workflows

Create Strategy

```
% Create the strategy
numAssets = size(pricesTT,2);
equalWeightsVector = ones(1,numAssets) / numAssets;
equalWeightsRebalanceFcn = @(~,~) equalWeightsVector;

ewStrategy = backtestStrategy("EqualWeighted",equalWeightsRebalanceFcn, ...
    'RebalanceFrequency',60, ...
    'LookbackWindow',0, ...
    'TransactionCosts',0.005, ...
    'InitialWeights',equalWeightsVector)
```

Run Backtest

```
% Create the backtest engine. The backtest engine properties that hold the
% results are initialized to empty.
backtester = backtestEngine(ewStrategy)
% Run the backtest. The empty properties are now populated with
% timetables of detailed backtest results.
backtester = runBacktest(backtester,pricesTT)
```

Backtest Summary

```
% Examining results. The summary table shows several performance metrics.
summary(backtester)
```

Backtesting ~2020

To interactive coding using live tasks

▼ **Backtest Portfolio Allocation** ▶ Autorun ? ⋮

`backtester` = Backtest strategies over market data pricesTT

Select data

Prices Signal

Define backtesting engine

Strategies Risk free rate Cash borrow rate

Rates convention Basis

Initial portfolio value Date adjustment

Run backtest for selected window

Start End

Backtesting ~2023

MATLAB and Python are both used in Finance

MATLAB



Python



A common question is asked:

Do I need to recode if I want to use MATLAB or Python?

MATLAB and Python work together

MATLAB can call Python



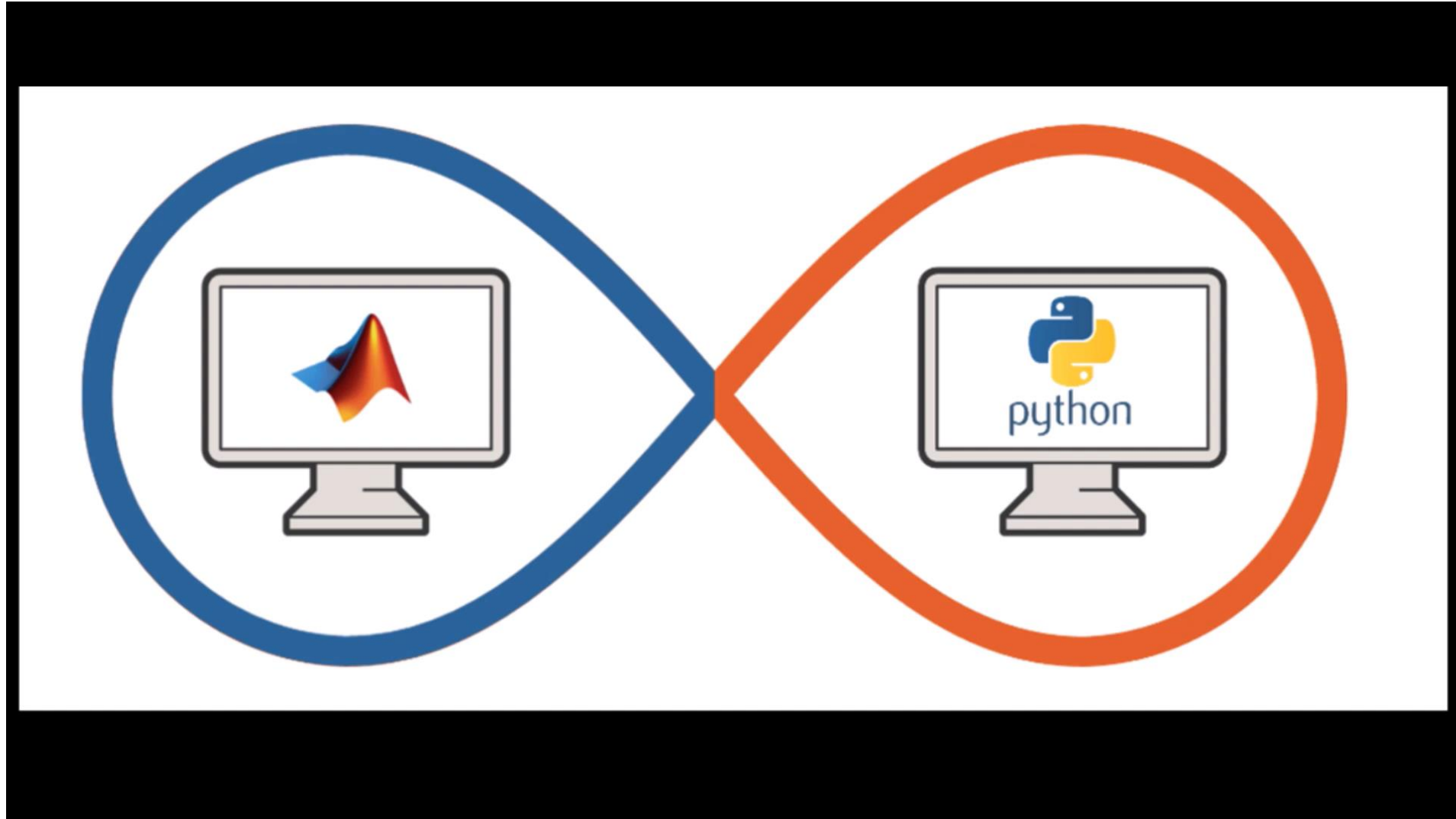
Python Interface

Python can call MATLAB



MATLAB Engine

Here is how MATLAB and Python can work together



Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

Automation = increased productivity
Interactive programming = increased productivity
MATLAB & Python = increased productivity

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

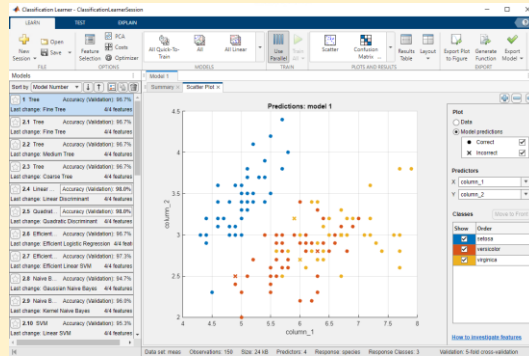
4. ModelOps revolutionizes model deployment and monitoring

Automation = increased productivity
Interactive programming = increased productivity
MATLAB & Python = increased productivity

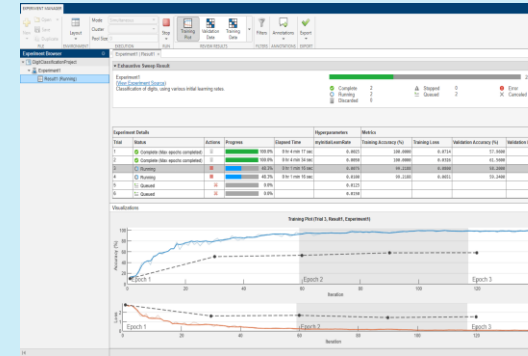
Trend #2

Low code / No code democratizes software development process

Updates to existing Low Code apps *Classification Learner*



New low code *Experiment Manager* Autogenerate live tasks



MY TASKS

Convert to Task

myBacktest

Live tasks can be auto-generated

The image shows the MATLAB R2023b Live Editor interface. The top menu bar includes HOME, PLOTS, APPS, LIVE EDITOR, INSERT, and VIEW. The ribbon contains various toolbars for FILE, NAVIGATE, TEXT, CODE, SECTION, and RUN. The current file is 'RefactorCreateBacktestStrategyLiveTask.mlx' located at 'C:\Users\dwilling\OneDrive - MathWorks\NAFAErious\Expo 2023\portfolio-backtesting-live-task'. The code editor shows the following code:

```
1 equalWeightFcn = @(w,p) ones(1,width(p)) / width(p);
2 initWeights = equalWeightFcn(0,pricesTT);

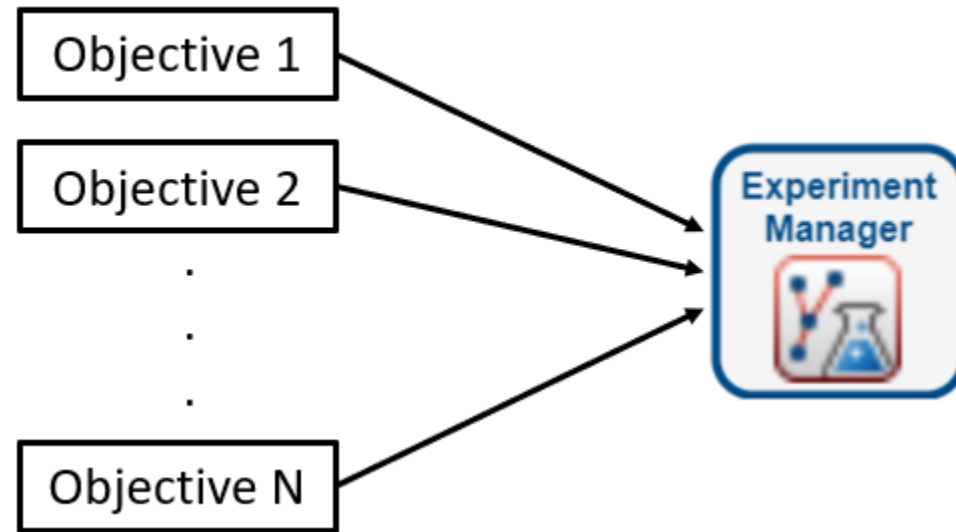
3 strategyName = NewStrategy; rebalanceFunction = equalWeightFcn;
4 rebalanceFrequency = 1; transactionCosts = 0;
5 lookbackWindow = 1; initialWeights = initWeights;
6 strategy = backtestStrategy(strategyName, rebalanceFunction, ...
7     "RebalanceFrequency", rebalanceFrequency, ...
8     "TransactionCosts", transactionCosts, ...
9     "InitialWeights", initialWeights, ...
10    "LookbackWindow", lookbackWindow);
```

Below the code, a live task configuration is shown:

```
strategy =
    backtestStrategy with properties:
        Name: "NewStrategy"
        RebalanceFcn: @(w,p)ones(1,width(p))/width(p)
```

The Command Window at the bottom shows the prompt `fx >>`. The status bar at the bottom indicates Zoom: 150%, UTF-8, LF, script, Ln 10, Col 37.

Experiment Manager can help find answers to multi-objective problems



Experiment Manager can help find answers to multi-objective problems

The screenshot displays the MATLAB Experiment Manager interface. The top menu bar includes HOME, PLOTS, APPS, LIVE EDITOR, INSERT, and VIEW. The ribbon contains various toolbars for file operations (New, Open, Save, Print, Export), navigation (Go To, Find, Bookmark), text formatting (Normal, Bold, Italic, Underline, Monospace), code editing (Code, Control, Task, Refactor), and execution (Run, Run and Advance, Run to End, Run, Step, Stop). The current folder is 'C:\Demos\managing-asset-allocation-with-experiment-manager\Demos'. The file list includes 'Data.xlsx', 'equalweight.m', 'erc.m', 'Experiment1_training1.mlx', 'Experiment1.mat', 'importdata.m', and 'mdp.m'. The main editor shows a 'Custom Training Experiment' section with the following text:

Custom Training Experiment

Use this training function to define the training data, network architecture, training options, and training procedure used by the experiment. Experiment Manager saves the output of this function, so you can export it to the MATLAB workspace when the training is complete. For more information, see [Configure Custom Training Experiment](#).

Input

- `params` is a structure with fields from the Experiment Manager hyperparameter table.
- `monitor` is an `experiments.Monitor` object that you can use to track the progress of the training, update information fields in the results table, record values of the metrics used by the training, and produce training plots.

Output

- `output` is the output returned from the training function.

The code editor shows the following function definition:

```
1 function output = Experiment1_training1(params,monitor)
Initialize custom metric functionality
2
```

The Command Window shows the prompt `>>`. The Workspace window shows a table with columns 'Name' and 'Value'.

Managing and Fine-Tuning Portfolio Optimization Workflows with Experiment Manager – Webinar Nov 14

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

Lower level of entry for new programmers
Enables all users to code quicker

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

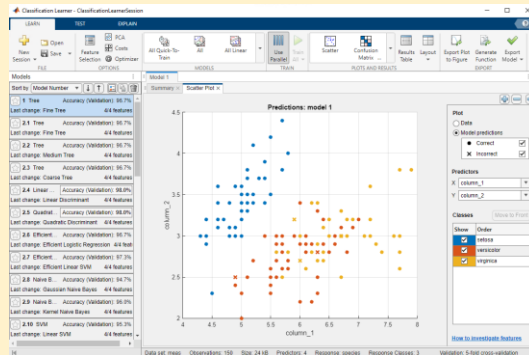
Lower level of entry for new programmers
Enables all users to code quicker

Trend #3

AI and Quantum improves existing systems

Existing examples of AI improvement

Explainability in Classification Learner



Emerging new technologies

ChatGPT/LLMs
Quantum*
Reinforcement Learning*

Multiperiod Goal-Based Wealth Management
Using Reinforcement Learning

Quantum Innovation in Finance: Portfolio
Optimization and Monte Carlo Simulation

Explain how a Machine Learning works

The screenshot shows the MATLAB R2023b Live Editor interface. The script 'importCreditData.m' contains the following code:

```
1 CreditData = importCreditData("CreditRatings.xlsx")
```

The output of the script is a table with 9 rows and 7 columns:

	ID	WC_TA	RE_TA	EBIT_TA	MVE_BVTD	S_TA
1	62394	0.0130	0.1040	0.0360	0.4470	0.1420
2	48608	0.2320	0.3350	0.0620	1.9690	0.2810
3	42444	0.3110	0.3670	0.0740	1.9350	0.3660
4	48631	0.1940	0.2630	0.0620	1.0170	0.2280
5	43768	0.1210	0.4130	0.0570	3.6470	0.4660
6	39255	-0.1170	-0.7990	0.0100	0.1790	0.0820
7	62236	0.0870	0.1580	0.0490	0.8160	0.3240
8	39354	0.0050	0.1810	0.0340	2.5970	0.3880
9	40326	0.4700	0.7520	0.0700	11.5960	1.1200

The Command Window at the bottom shows the prompt `f> >>`. The status bar at the bottom indicates Zoom: 150%, UTF-8, LF, and script.

AI Chat playground – your MATLAB coding assistant

The screenshot shows the MATLAB AI Chat Playground interface in a web browser. The browser's address bar shows the URL `mathworks.com/matlabcentral/playground/new`. The page header includes the MathWorks logo and navigation links for Products, Solutions, Academia, Support, Community, and Events. A blue banner at the top of the page reads "AI Chat Playground".

On the left side, there is a "Clear" button and a section for the AI assistant. The AI's message reads: "Hello! Whether you're checking out MATLAB for the first time or have been using it for years, I'm here to answer your questions and provide coding tips." Below this is a warning box: "Please keep in mind that AI sometimes writes code and text that seems accurate, but isn't. AI does not yet have knowledge of features delivered after June 2021 and only limited knowledge of Simulink and specialized toolboxes. This is a space for experimentation. Try it, verify any resulting code, and kindly give thumbs up or thumbs down on the results to help improve the responses." There are also two example prompts: "Highlight contours at particular levels" and "Create a vertical line at x = 3". A "Shuffle" button is located next to the second prompt. Below the prompts is a text input field with the placeholder "Send a message..." and a send button.

On the right side, there is a code editor area with a toolbar containing icons for zooming, undo, redo, and running code. The "CODE" and "RUN" buttons are visible. The code editor itself is currently empty.

At the bottom of the page, there is a small disclaimer: "Generated code and text might be inaccurate. Validate before use. [About](#)".

AI Chat playground – sign up for early access



Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

AI = improving existing systems
Quantum = Faster computation

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

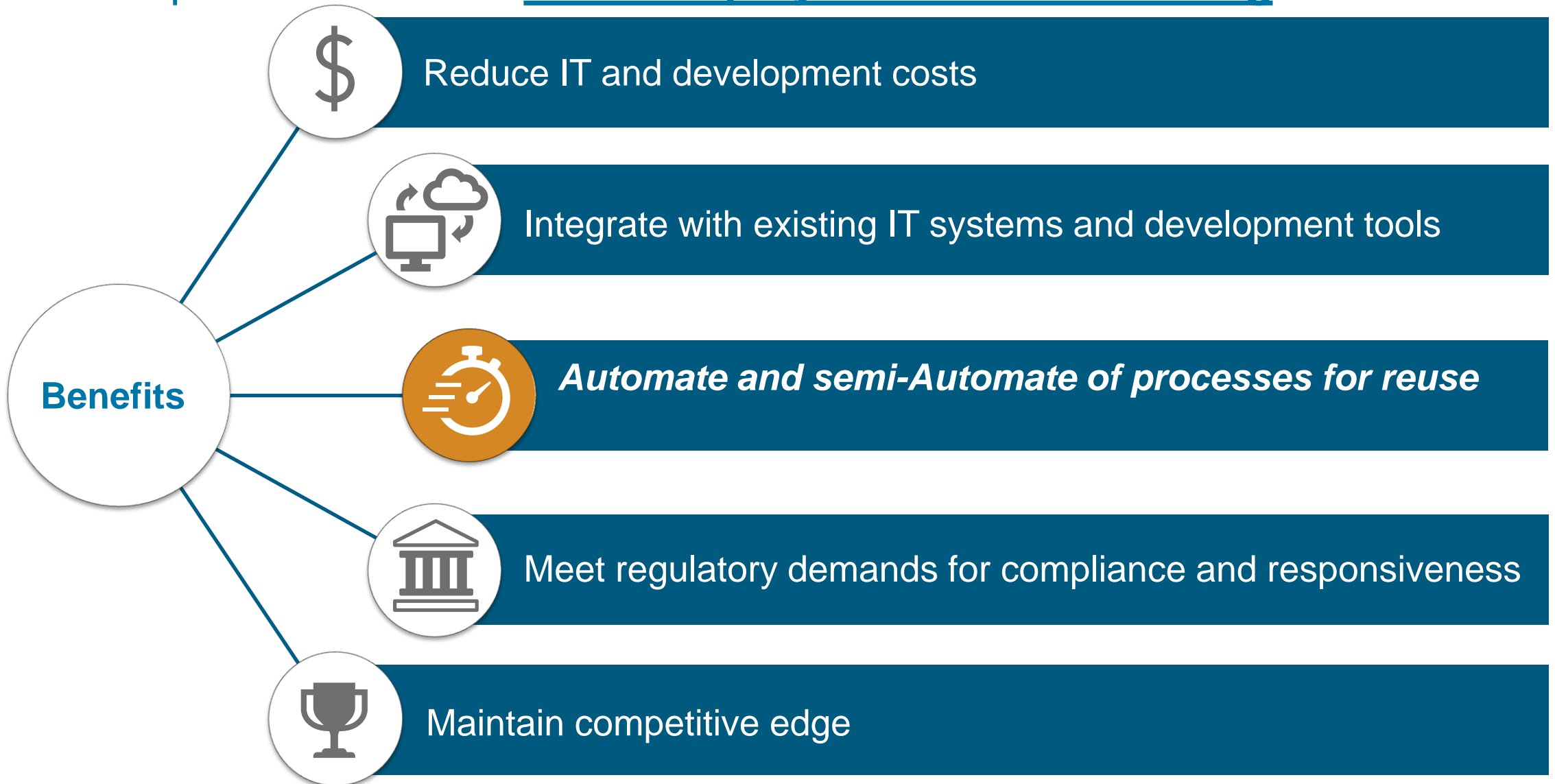
3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

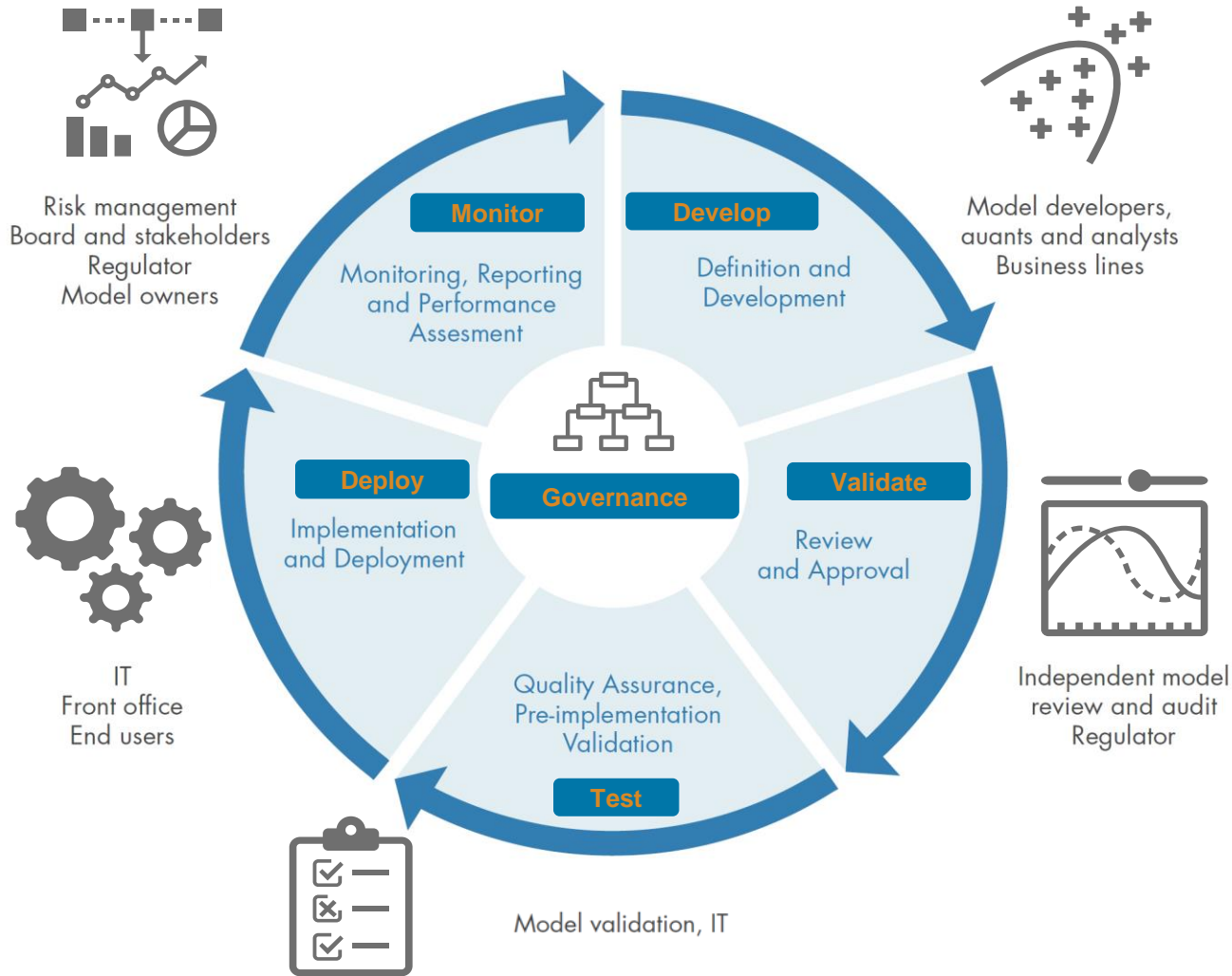
AI = improving existing systems
Quantum = Faster computation

Trend #4

ModelOps revolutionizes model deployment and monitoring



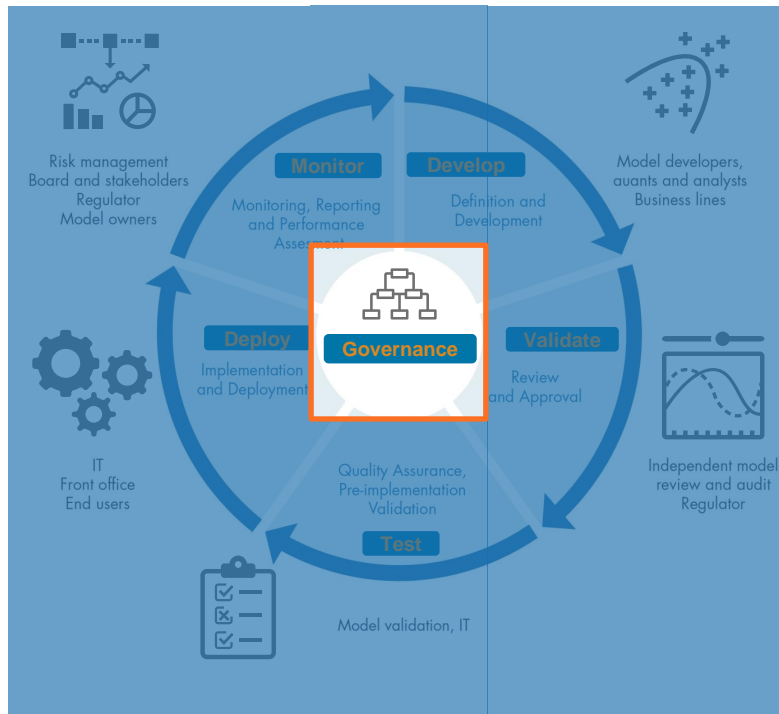
MathWorks has a solution for the entire model lifecycle workflow



Modelscope

Many organizations are new to Model Lifecycle Management

Model Governance is their first entry point



MODEL RISK MANAGEMENT

ModelOps helps streamline Model and App deployment

The screenshot displays the MATLAB Web App Server interface. At the top, there is a navigation bar with the text "MATLAB Web App Server" and a "Diagnostics" button. Below this, the page is organized into sections. The "Asset Allocation" section features four application cards: "ESG High Yield Portfolio Optimization - Replication Strategy" (version 1.0), "ESGScoreOptimizeAPP" (version 1.0), "High Yield Portfolio Optimization - Replication Strategy" (version 1.0), and "HRP_App" (version 1.0). The "Client - Private" section features three application cards: "Business Responsibility and Sustainability Report" (version 2.0), "FRTB_SA_SBM" (version 1.0), and "Mixed Integer Quadratic Portfolio Optimization" (version 1.0). Each card includes a thumbnail image, the application title, the author's name, a brief description, and the version number. A dark blue banner at the bottom of the screenshot contains the URL <https://xlab.mathworkscloud.com/webapps/home/>. The word "Econometrics" is visible in the bottom left corner of the screenshot.

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

ModelOps = Automate & Semi-Automate model development, deployment and monitoring

Market trends driving Quantitative Finance

Impact on MATLAB Users

1. Automation streamlines existing manual tasks and processes

Automation = increased productivity
Interactive programming = increased productivity
MATLAB & Python = increased productivity

2. Low code / No code democratizes the software development process

Lower level of entry for new programmers
Enables all users to code quicker

3. AI and Quantum improves existing systems

AI = improving existing systems
Quantum = Faster computation

4. ModelOps revolutionizes model deployment and monitoring

ModelOps = Automate & Semi-Automate
model development, deployment and
monitoring

Our sessions Day 1

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

Talks

Multiperiod Goal-Based Wealth Management Using Reinforcement Learning



Development of a Performance Analysis App from Design to Deployment



Extending the Scope: From Back-Office Engine to Growing Front-Office Platform



Quantum Innovation in Finance: Portfolio Optimization and Monte Carlo Simulation



Review of AI and Machine Learning Usage in Financial Applications



Our sessions Day 2

1. Automation streamlines existing manual tasks and processes

2. Low code / No code democratizes the software development process

3. AI and Quantum improves existing systems

4. ModelOps revolutionizes model deployment and monitoring

Talks

Modeling the Impact of Climate Change on Insured Losses in France



CRISK: Quantifying the Expected Capital Shortfall in a Climate Stress Scenario



Dynare: Macroeconomic Modeling for All



BEAR Toolbox for Estimating Economic Relationships



Nonlinear Confidence Bands Computation in MATLAB



Foreign Economic Policy Uncertainty and US Equity Returns



