Streamlining Financial Modelling: From Development to Approval to Production with MATLAB

David Sampson
MathWorks
Landscape
- Requirements
- Challenges

Technology
- Environment
- Toolboxes
- Integrations

Workflows
- How they work
- Why they work
Landscape
• Requirements
• Challenges

Technology
• Environment
• Toolboxes
• Integrations

Workflows
• How they work
• Why they work
Model risk management regulations

2000
OCC 2000-16
First regulatory definition of models and risk

2006
CEBS GL10
Basel 2, new validation requirements

2011
OCC/FED SR11-7
First comprehensive supervisory guidance on model risk management

2013-14
CRDIV – CRR
EBA SREP
CP/2014/14
Model risk integrated as part of Pillar 2 (Europe)

2016
TRIM
RTS2016/03
3 Lines of Defence
Risk Management Framework Development Review
Regulator requirements

quantitative, qualitative, competence, roles, limitations, judgement, uncertainty, inventory, understand, thirdparty, data, sensitivity, process, proportionate, responsibilities, transparency, models, impact, verify, credibility, external, independence.
Institution challenges

- Definition
- Development
- Validation
- Approval
- Deployment
- Monitoring

- Consistency
- Segmentation
- Availability
- Usability
- Sensitivity
- Recoding
- Security
- Insight
- Templates
- Throughput
- Standards
- Traceability
- Interoperability
- Integration
- Range
- Effort
- Model
- Robustness
- Reuse
- Data
- Review
- External
- Validity
- Readability
- Rework
Landscape
- Requirements
- Challenges

Technology
- Environment
- Toolboxes
- Integrations

Workflows
- How they work
- Why they work
In the beginning...

Read more at Cleve's Corner
Desktop
function g = i_func( f, uniformOutput, varargin ) %#ok<INUSL>

% i_func Create function handle in local workspace.

% i_func(f,uo,a,b,...) with uo true (uniform output) creates the function
% handle @(ii)f(a(ii),b(ii),...). This is used when f returns a scalar
% per group.
% i_func(f,uo,a,b,...) with uo false (nonuniform output) creates the
% function handle @(ii){f(a(ii),b(ii),...)}. This is used when f returns
% a nonscalar per group.

% Create construction string for function handle
vars = sprintf( 'a%d = varargin%d;', repmat( 1:numel( varargin ), [2 1] ) );
eval( vars );
args = sprintf( 'a%d(ii)', 1:numel( varargin ) );
if uniformOutput % return output directly
    func = sprintf( '@(ii)f(%s);', args(1:end-1) );
else % return output in a cell
    func = sprintf( '@(ii){f(%s)};', args(1:end-1) );
end
## Types of MATLAB Plots

There are various functions that you can use to plot data in MATLAB®. This table classifies and illustrates the common graphics functions.

<table>
<thead>
<tr>
<th>Line Plots</th>
<th>Pie Charts, Bar Plots, and Histograms</th>
<th>Discrete Data Plots</th>
<th>Polar Plots</th>
<th>Contour Plots</th>
<th>Vector Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>plot</code></td>
<td><code>area</code></td>
<td><code>stairs</code></td>
<td><code>polarplot</code></td>
<td><code>contour</code></td>
<td><code>quiver</code></td>
</tr>
<tr>
<td><code>plot3</code></td>
<td><code>pie</code></td>
<td><code>stem</code></td>
<td><code>polarhistogram</code></td>
<td><code>contourf</code></td>
<td><code>quiver3</code></td>
</tr>
<tr>
<td><code>semilogx</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>semilogy</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example

![MATLAB Graphics Example](image-url)

- **Generate code!**

---

**Documentation**

[Search Help]
Toolboxes

Econometrics Toolbox

Model and analyze financial and economic systems using statistical methods.

Econometrics Toolbox™ provides functions for modeling economic data. You can select and estimate economic models for simulation and forecasting. For time series modeling and analysis, the toolbox includes univariate Bayesian linear regression, univariate ARIMAX/GARCH composite models with several GARCH variants, multivariate VARX models, and cointegration analysis. It also provides methods for modeling economic systems using state-space models and for estimating using the Kalman filter. You can use a variety of diagnostics for model selection, including hypothesis tests, unit root, stationarity, and structural change.

**Getting Started**
Learn the basics of Econometrics Toolbox.

**Data Preprocessing**
Format, plot, and transform time series data.

**Model Selection**
Specification testing and model assessment.

**Time Series Regression Models**
Bayesian linear regression models and regression models with nonspherical disturbances.

**Conditional Mean Models**
Autoregressive (AR), moving average (MA), ARMA, ARIMA, ARIMAX, and seasonal models.

**Conditional Variance Models**
GARCH, exponential GARCH (EGARCH), and GJR models.

**Multivariate Models**
Cointegration analysis, and vector autoregressive (VAR) and vector error correction (VEC) models.
Documentation Browser

**fmincon**

Find minimum of constrained nonlinear multivariable function

Nonlinear programming solver

Finds the minimum of a problem specified by

\[
\begin{align*}
    \min f(x) & \quad \text{such that} \\
    c(x) & \leq 0 \\
    ceq(x) & = 0 \\
    A \cdot x & \leq b \\
    Aeq \cdot x & = beq \\
    lb \leq x & \leq ub,
\end{align*}
\]

A and beq are matrices. \(A\) and \(lb\) are vectors. \(\text{fun}, \text{A}, \text{Aeq}, \text{lb}, \text{ub}, \text{nonlcon}\) are functions that return vectors, and \(\text{f}<>\text{fun}\) is a function that returns a scalar. \(\text{lb}\) and \(\text{ub}\) can be passed as vectors or matrices; see **Matrix Arguments**.

**Syntax**

\[
\begin{align*}
    x &= \text{fmincon}(<\text{fun}, \text{A}, \text{b}, \text{Aeq}, \text{beq}, <\text{lb}, \text{ub}, \text{nonlcon}>, \text{options}>), \\
    [x, \text{fval}] &= \text{fmincon}(<\text{fun}, \text{A}, \text{b}, \text{Aeq}, \text{beq}, <\text{lb}, \text{ub}, \text{nonlcon}, \text{options}>)
\end{align*}
\]

**Description**

\(x = \text{fmincon}(<\text{fun}, \text{A}, \text{b}, \text{Aeq}, \text{beq}>)\) starts at \(x0\) and attempts to find a minimizer \(x\) of the function described in \(\text{fun}\) subject to the matrix constraints.

**Note**

**Passing Extra Parameters** explains how to pass extra parameters to the objective function and nonlinear constraint functions.
Variable Editor

```
Command Window

>> TMW

TMW =

1000x5 timetable

<table>
<thead>
<tr>
<th>Time</th>
<th>Open</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-Sep-2012</td>
<td>100</td>
<td>102.19</td>
</tr>
<tr>
<td>05-Sep-2012</td>
<td>100.15</td>
<td>101.05</td>
</tr>
<tr>
<td>06-Sep-2012</td>
<td>100.4</td>
<td>102.38</td>
</tr>
<tr>
<td>07-Sep-2012</td>
<td>101.74</td>
<td>102.37</td>
</tr>
<tr>
<td>10-Sep-2012</td>
<td>99.72</td>
<td>101.55</td>
</tr>
<tr>
<td>11-Sep-2012</td>
<td>98.48</td>
<td>98.66</td>
</tr>
<tr>
<td>12-Sep-2012</td>
<td>96.9</td>
<td>99.18</td>
</tr>
</tbody>
</table>
```
Live Editor

Use Interactive Controls in a Live Script

This example shows how you can add interactive controls to a live script. Adding interactive controls to a live script is useful when you want to share the live script with others. Use interactive controls to set and change the values of variables in one live script or familiar user interface components such as numeric sliders and drop-down lists.

Add Interactive Controls to Your Script

To add an interactive control, select a value in a variable declaration, and select either *Numeric Slider* or *Dropdown* to configure the interactive control. Double-click it when you can select the section that contains the control.

Visualize Airport and Carrier Delays

Use the interactive controls in the live script to change the values of the delay parameters and the top carriers. The plotDelays function creates bar charts to display the delays for the top airports.

Average Delay by Airport in 1999

Average Delay by Carrier in 1999

18
generate code!
Toolbox Packaging

This toolbox provides tools to create sophisticated MATLAB graphical user interfaces that resize used in combination to produce virtually any user interface layout.

* Arrange MATLAB user interface components horizontally, vertically or in grids

**Toolbox Files and Folders**
- layout
- layoutdoc
Report Generator

Documentation

CONTENTS

Create Presentation Content

Use the MATLAB® API for PowerPoint® (PPT API) to create presentation content.

Use the PPT API to create MATLAB programs to add content to PowerPoint presentations and to create complete PowerPoint presentations. You can create a program that works with the slide master and layouts in a PowerPoint presentation. To get started, see Create a Presentation Generator.

To share your completed presentation program with others who do not have MATLAB installed on their systems, see Compile a Presentation Program.

Functions

- `mrptgen ppt.Presentation.open`: Open presentation.
- `mrptgen ppt.Presentation.close`: Close presentation.
- `mrptgen ppt.Presentation.getMasterNames`: Get names of slide masters for presentation.
- `mrptgen ppt.Presentation.getLayoutNames`: Get names of layouts for presentation slide master.
- `mrptgen ppt.Presentation.getTablestyles`: Get table style names for presentation.
- `mrptgen ppt.Presentation.add`: Add slide to presentation.
- `mrptgen ppt.Presentation.replace`: Replace paragraphs, titles, or pictures in presentation.
- `mrptgen ppt.ProgressMessage.formatAsText`: Format message as text.
- `mrptgen ppt.ProgressMessage.formatAsHTML`: Wrap message in HTML tags.
- `mrptgen ppt.ProgressMessage.passFilter`: Determine if message passes filter.
- `pptShow`: Open Microsoft PowerPoint presentation or convert it to PDF.

Classes

- `mrptgen ppt.Presentation`: Create Microsoft PowerPoint presentation.
- `mrptgen ppt.slide`: Presentation slide.
Parallel

- High-level: parfor, gpuArray
- Low-level: batch, createJob, createTask
- Big data: tall, mapreduce
Enterprise integration

Access and Explore Data

Preprocess Data

Develop Predictive Models

Integrate Analytics with Systems

Databases
- DynamoDB
- Microsoft SQL Server
- MongoDB
- Cassandra
- Cosmos DB

Cloud Storage
- Azure Blob

Big Data / OT
- cloudera
- HORTONWORKS
- OSIsoft, PI System

Streaming
- AWS Kinesis
- Azure IoT Hub

OT Platforms
- kafka

Dashboards
- Qlik Sense
- Microsoft Power BI
- Spotfire

Matlab
- MDCS
- MATLAB Production Server
- Request Broker
Landscape
• Requirements
• Challenges

Technology
• Environment
• Toolboxes
• Integrations

Workflows
• How they work
• Why they work
Integrate Analytics with Systems

Preprocess Data

Develop Predictive Models

Integrate Analytics with Systems

Files

Working with Messy Data

Model Creation e.g. Machine Learning

Desktop Apps

Databases

Data Reduction/Transformation

Parameter Optimization

Enterprise Scale Systems

Sensors

Feature Extraction

Model Validation

Embedded Devices and Hardware

Access and Explore Data

Files

Databases

Sensors

Working with Messy Data

Data Reduction/Transformation

Feature Extraction

Model Creation e.g. Machine Learning

Parameter Optimization

MATLAB, Excel, .NET, C/C++, Java, dll

Desktop Apps

Enterprise Scale Systems

Embedded Devices and Hardware

Files

Databases

Sensors

Working with Messy Data

Data Reduction/Transformation

Feature Extraction

Model Creation e.g. Machine Learning

Parameter Optimization

MATLAB, Excel, .NET, C/C++, Java, dll

Desktop Apps

Enterprise Scale Systems

Embedded Devices and Hardware

Files

Databases

Sensors

Working with Messy Data

Data Reduction/Transformation

Feature Extraction

Model Creation e.g. Machine Learning

Parameter Optimization

MATLAB, Excel, .NET, C/C++, Java, dll

Desktop Apps

Enterprise Scale Systems

Embedded Devices and Hardware
Data preparation

Assemble a sufficient volume of clean data of known provenance.

>> importdata
Model generation

Create models with the required **accuracy** and **reach**, while driving **insight**.
Generate an accurate, insightful description of the study in a timely manner.
Model review

Provide an **accurate, thorough** view of the study that allows others to **engage**.
Model deployment

Provide approved, accurate, current models for use throughout the business.