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

Definition

Framework

Development

Review
Regulator requirements

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

consistency, validity, availability, segmentation, usability, sensitivity, recoding, data, review, insight, effort, range, internal, templates, throughput, standards, integration, traceability, interoperability
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
    func = sprintf( '
        @(ii)f(%s);', args(1:end-1) );
else
    func = sprintf( '
        @(ii){f(%s)};', args(1:end-1) );
end
Graphics

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 Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>plot</td>
<td>area</td>
<td>stairs</td>
<td>polarplot</td>
<td>contour</td>
<td>quiver</td>
</tr>
<tr>
<td>plot3</td>
<td>pie</td>
<td>stem</td>
<td>polarhistogram</td>
<td>contourf</td>
<td>quiver3</td>
</tr>
</tbody>
</table>

Generate code!
**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**

**Contents**
- My Products
- MATLAB
- Simulink
- Aerospace Blockset
- Aerospace Toolbox
- Audio System Toolbox
- Automated Driving System Toolbox
- Bioinformatics Toolbox
- Communications System Toolbox
- Computer Vision System Toolbox
- Control System Toolbox
- Curve Fitting Toolbox
- Data Acquisition Toolbox
- Database Toolbox
- Debugging Toolbox
- DOPPLER Toolbox
- DSP System Toolbox
- Econometrics Toolbox
- Embedded Coder
- Financial Instrument Toolbox
- Financial Toolbox
- Fixed-Point Designer
- Fuzzy Logic Toolbox
- Global Optimization Toolbox
- GPU Coder
- HDL Coder
- IEC Certification Kit (for ISO 26262 and IEC 61508)
- Image Acquisition Toolbox
- Image Processing Toolbox
- Instrument Control Toolbox
- Interface Control Documentations
- Intelligent Systems
- Kalman Filter
- LTI System
- Mapping Toolbox
- MATLAB
- MATLAB® Language
- Math, Statistics, and Optimization
- MathWorks
- Partial Differential Equation Toolbox
- Parallel Computing Toolbox
- Parallel Table Toolbox
- Parallel Computing
- Partial Differential
- Statistics, and Machine Learning Toolbox
- Symbolic Math Toolbox
- Text Analytics Toolbox
- Test 

**Examples**
- MATLAB Examples
  - Language Fundamentals
  - Basic Matrix Operations
  - Matrix Manipulation
  - Manipulating Multidimensional Arrays
- Create Structure
- Create a structure with fields and data using data structre.
- Mathematics
- Calculations
- Integer Arithmetic
- Single Precision Math
- Creating and Editing Delaunay Triangulations
- Predicting the US Population
- FFT for Spectral Analysis

**Documentation**

**Contents**
- fmincon

Find minimum of constrained nonlinear multivariable function

Nonlinear programming solver

Finds the minimum of a problem specified by

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

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

**Syntax**

```matlab
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
x = fmincon(problem)
[x,fval] = fmincon(__)
[x,fval,exitflag,output] = fmincon(__)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(__)
```

**Description**

- `fmincon` starts at \( \text{x0} \) and attempts to find a minimizer \( \text{x} \) of the function described in `fun` subject to the nonlinear constraints defined by fun.

**Note**

- See `Passing Extra Parameters` on how to pass extra parameters to the objective function and nonlinear constraint functions.

- \( x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon) \) minimizes `fun` subject to the linear equalities `Aeq` \( x = \text{beq} \) and \( \text{A}x \leq b \). If no bounds are given, \( x \) is unconstrained.

- \( x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon) \) defines a set of lower and upper bounds on the design variables in \( \text{x} \). If `lb` or `ub` is `[]`, then that `x` component is unconstrained.
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>
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 code and other 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 code and change the variable name to an interactive variable name. Then, select the interactive variable in the section that contains the control.

Visualize Airport and Carrier Delays

Use the interactive controls in the live script to create and customize plots. This can help you see the relationship between multiple delays and the top carriers. The `plotDelays` function creates bar charts to display the delays for the top airports and top carriers. The `plotDelays` function creates bar charts to display the delays for the top airports and top carriers.

Air Mass and Solar Radiation

<table>
<thead>
<tr>
<th>Air Mass</th>
<th>Solar Radiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Equation</td>
</tr>
</tbody>
</table>

The larger the air mass, the less radiation reaches the ground. The air mass can be calculated from the equation:

\[ AM = \frac{f}{v(10-v)} = \frac{0.000005}{0.60799 + \frac{v}{10}} \]

Then the solar radiation (in W/m²) reaching the ground can be calculated from the empirical equation:

\[ s_{rad} = 1.353 + 0.815 AM \]
Apps

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

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

- `reportgen.ppt.Presentation.open` Open presentation
- `reportgen.ppt.Presentation.close` Close presentation
- `reportgen.ppt.Presentation.getMasterNames` Get names of slide masters for presentation
- `reportgen.ppt.Presentation.getLayoutNames` Get names of layouts for presentation slide master
- `reportgen.ppt.Presentation.getTablestylesNames` Get table style names for presentation
- `reportgen.ppt.Presentation.add` Add slide to presentation
- `reportgen.ppt.Presentation.replace` Replace paragraphs, titles, or pictures in presentation
- `reportgen.ppt.MessageDispatcher.dispatch` Dispatch PPT status message
- `reportgen.ppt.MessageDispatcher.getMessageDispatcher` Return PPT message dispatcher
- `reportgen.ppt.ProgressMessage.formatAsText` Format message as text
- `reportgen.ppt.ProgressMessage.formatAsHTML` Wrap message in HTML tags
- `reportgen.ppt.ProgressMessage.passOrFilter` Determine if message passes filter
- `pptview` Open Microsoft PowerPoint presentation or convert it to PDF

Classes

- `reportgen.ppt.Presentation` Create Microsoft PowerPoint presentation
- `reportgen.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
- 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
- OSIsoft, PI System

Dashboards
- Qlik
- Microsoft Power BI
- Spotfire

MATLAB
MDCS
Request Broker

Cloud Storage
Azure

Big Data / OT
cloudera

Streaming
AWS Kinesis
Azure IoT Hub

OT Platforms
OSIsoft, PI System

Dashboards
Qlik
Microsoft Power BI
Spotfire
Workflows

- How they work
- Why they work

Technology

- Environment
- Toolboxes
- Integrations

Landscape

- Requirements
- Challenges
Assemble a sufficient **volume** of **clean** data of known provenance.
Create models with the required accuracy and reach, while driving insight.
Generate an **accurate**, **insightful** description of the study in a **timely** manner.
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.