Trainings Objectives

This course part will be splitted into the topic «Data Processing» and the «visualization» aspect discussed under the topic «Building Graphical User Interfaces»

MATLAB for Data Processing focuses on the details of data management. The course emphasizes creating scripts that extend the basic features provided by MATLAB. Hands-on examples explore features for efficiently organizing and presenting data, providing a practical set of tools for further data analysis. Topics include:

- Importing data
- Organizing data
- Exporting data

Day 1 of 5

Importing Data

Objective: Data files come in many formats, from spreadsheets to plain text. This first section of the course dissects the many facets of reading files of various types and formats. Emphasis is given to irregular text files, which contain a mixture of data types, delimiters, and headers.

- File types and formats
- Interactive import methods
- Programmatic import methods
- Low-level import methods
- Importing a mixture of data types from text files using textscan
- Large data sets and irregular formats
- Batch import tasks

Organizing Data

Objective: There are many ways to store data in the MATLAB environment. This section explores the tradeoffs involved in choosing an appropriate data type, highlighting the built in data types of cell arrays and structures. Additionally, this section demonstrates some of the common techniques for organizing and processing data.

- Cell arrays
- Set operations
- Function handles
- Applying functions to an array
- Structure arrays
- Extracting and aggregating data
- Finding and counting

Exporting Data

Objective: A final stage of the of a successful data analysis includes the publication of results. Whether publishing a presentation, a report, or a simple text file for further analysis, this section will help in outputting results in various formats. Additional guidance in publishing to various graphical formats is also a feature of this section.
MATLAB - Workshop

- Writing numeric data to text files
- Writing a mixture of data types to text files
- Saving data to other file formats
- Exporting and printing graphics
- Saving animations
- Publishing

Part 2 MATLAB Programming Techniques

Training Objectives

MATLAB Programming Techniques provides hands-on experience using the features in the MATLAB language to write efficient, robust, and well-organized code. These concepts form the foundation for writing full applications, developing algorithms, and extending built-in MATLAB capabilities. Details of performance optimization are covered, as well as tools for writing, debugging, and profiling code. Topics include:

- Programming for correctness
- Structuring data
- Structuring code

Course Outline

Day 1 and 2 of 5

Programming for Correctness

Objective: Creating robust applications that withstand unexpected input and produce meaningful errors is the goal of many application developers. Through the use of built-in MATLAB functions and programming constructs, this section covers standard techniques for handling error conditions. Tools for debugging code, diagnosing problems, and measuring performance are an integral part of this section.

- Warnings and errors
- Handling errors
- The try-catch construct
- The MException object
- M-Lint code check
- Directory reports
- MATLAB Debugger
- Assessing performance
- MATLAB Profiler

Structuring Data

Objective: This section examines choices for storing data within a MATLAB application. The choice of data type plays an important role in an application's storage requirements and execution time. Highlighted in this section are common techniques for improving performance when storing, accessing, and processing data.

- Cell arrays
- Structure arrays
- Other data containers
MATLAB - Workshop

- Preallocation
- Vectorization
- Memory management
- Passing data between functions

**Structuring Code**

**Objective:** Modularizing code into readable and maintainable functions is often an important design consideration. This section explores the different function types available in the MATLAB language, and discusses some of the tradeoffs associated with these choices. The effect of the function type on performance, as well as reliability of the code, is an important theme throughout the chapter.

- Private functions
- Function handles
- Subfunctions
- Nested functions
- Precedence rules
- Variable scope and visibility
- Comparison of function types

**Part 4  Image Processing with MATLAB**

**Training Objectives**

This part shows how to use Image Processing Toolbox™ to perform various image processing techniques. The course explores the different types of image representations; as well as how to enhance image characteristics, filter and image, and reduce the effects of noise and blurring in an image. It also introduces different methods used to extract features and objects within an image, image registration, and techniques for reconstructing images and objects.

Basic knowledge of image processing concepts is strongly recommended.

**Course Outline**

**Day 2 and 3 of 5**

**Working with Images**

**Objective:** Understand different image types available in MATLAB, and how they can be read into MATLAB.

- Image types
- Supported MATLAB data types for representing images
- Binary images
- Grayscale images
- Indexed images
- RGB images
- Importing and exporting images in MATLAB
- Viewing images
- Single image
- Multiple image frames
- Finding image pixel values
- Calculating image statistics
MATLAB - Workshop

- Converting image formats

**Image Enhancement Techniques**

**Objective:** Enhance image characteristics by adjusting the image intensity and isolating a region of interest.

- Adjusting image intensity
- Histogram stretching
- Histogram equalization
- Histogram adjustment
- Using arithmetic functions to enhance images
- Correcting image alignment: rotating
- Cropping and resizing images
- Exploring the basics of image registration
- Selecting control points
- Registering an image
- Correcting lens distortion

**Filtering Images**

**Objective:** Understand how block processing works; implement spatial-domain and frequency-domain filters; and use filtering techniques to reduce the effects of unwanted distortions such as noise, blurring, and background illumination or to enhance an image.

- Defining filtering
- Filtering process
- Performing filtering
- Filtering applications: smoothing, edge detection, and sharpening
- Frequency-domain filter design
- Modeling and removing noise
- General block operations
- Region-of-interest processing
- Specific applications of filtering

**Day 3 of 5**

**Feature Extraction and Segmentation**

**Objective:** Extract image features and measurements using different segmentation methodologies.

- Isolating image features using thresholding
- Performing morphological segmentation
- Creation of structuring elements
- Erosion and dilation
- Measurement of region properties
- Reconstructing images and objects
- Performing morphological reconstruction
- Detecting edges in an image
- Hough transform
- Applying color-based image segmentation
- Isolating objects using watershed segmentation
- Segmenting images based on texture

**Appendix: Radon and Inverse Radon Transforms (optional)**
MATLAB - Workshop

Objective: Use the Radon and inverse Radon transforms to assist with detecting lines and viewing an output that looks similar to an original image.

Please note, that we have to skip parts of the topics above in case we should discuss aspects of this appendix.

- What is the Radon transform?
- What is the inverse Radon transform?
- How are these transforms used?
- What applications use these transforms?

Appendix: Wavelets (optional)

Objective: Use wavelets analysis for detecting features in images.

Please note, that we have to skip parts of the topics above in case we should discuss aspects of this appendix.

- Performing wavelets analysis using Wavelet Toolbox
- What are wavelets?
- Comparing Fourier and wavelet transforms
- Relationship between frequency and scale
- Detail and approximation
- Continuous and discrete wavelet transforms
- Decomposition and packet analysis
- Wavelet reconstruction
- Hard and soft thresholding
- Using wavelets with images
- Decomposition of 2-D signals
- Wavelets applications
- Example

Appendix: Large Data Handling (optional)

Objective: Import and process large images.

Please note, that we have to skip parts of the topics above in case we should discuss aspects of this appendix.

- What is large image data?
- Why is it important to handle large image data?
- How can large data be handled?

Part 5 MATLAB for Building Graphical User Interfaces and Visualization

Training Objectives

The Visualization part: See remarks Part 1.

The chapter about visualization focuses on the details of visualization techniques to producing customized publication-quality graphics.
The rest of the day will provide an introduction to building a graphical user interface (GUI) in MATLAB. Focus is on user interface controls, such as push buttons and text boxes, and on how to interact with them to create a robust GUI. Topics include:

- Visualization
- Handle Graphics
- User interface controls
- Callback functions
- Graphical user interface development environment (GUIDE)
- GUI deployment

Day 4 of 5

Handle Graphics®

Objective: Review Handle Graphics concepts and manipulate graphics objects programmatically.

- Graphics objects
- Object handles
- Property name / property value pairs
- Object hierarchy
- Current axes and figures

Visualizing Data

Objective: A good visualization can effectively communicate the results of an analysis. Using the plotting capabilities of MATLAB, this section aims to investigate many different techniques for presenting data. Highlighted are techniques for annotating and modifying standard plots into publication-quality graphics.

- Choosing a plot type
- Using color
- Customizing plots
- Handle Graphics® revisited
- Common plot applications
- Creating animations
- Writing user-defined plot functions

Components of a GUI Application

Objective: Create functional user interface controls on a MATLAB figure window.

- uicontrol objects
- Object position
- Callbacks
- Event execution
- Edit boxes

Programming Considerations for GUI Applications

Objective: Apply programming techniques to build robust, self-contained GUIs, and to organize GUI data.

- Function callbacks
- Local GUI data
MATLAB - Workshop

- Data organization
- Callback data

**Layout of GUI Applications Using GUIDE**

**Objective:** Use GUIDE to interactively design a graphical user interface.

- GUIDE workflow
- GUIDE layout
- Object properties
- GUIDE code template

**Programming GUI Applications Using the GUIDE Template**

**Objective:** Modify the callback function stubs created by GUIDE to create a fully functional GUI.

- GUI data
- GUI data additions and modifications
- Wrapper GUI

---

**Part 6  Statistical Methods in MATLAB**

**Training Objectives**

This part course provides an introduction to statistical tools in MATLAB and Statistics Toolbox, including:

- Importing and organizing data
- Computing descriptive statistics
- Visualizing data
- Generating random numbers and performing simulations
- Fitting distributions to data
- Performing bivariate and multivariate regression

**Course Outline**

**Day 5 of 5**

**Data Management**

**Objective:** Understand the import methods and data types available in MATLAB and Statistics Toolbox in order to bring data into MATLAB and organize it for analysis. Dealing with common problems, such as missing data, are highlighted.

- Importing data
- Organizing data
- Categorical data and dataset arrays
- Incommensurate and missing data

**Exploring Data**

**Objective:** Understand basic statistical investigation of a data set, including visualization and calculation of summary statistics.

- Descriptive statistics
- Central tendency
- Spread
Distributions and Random Numbers

Objective: Understand the functionality in Statistics Toolbox for investigating different probability distributions and generating random numbers from one of these or any other distribution.

- Probability distributions
- Distributions in Statistics Toolbox
- Generating random numbers
- Random number streams
- Random numbers for arbitrary distributions
- Monte Carlo simulation

Fitting and Testing Distributions

Objective: Use Statistics Toolbox to explore a data set and compare the data to a theoretical distribution, either to estimate parameters of the data's distribution or to test a hypothesis about the data.

- Choosing a distribution
- Fitting a distribution
- Testing a distribution
- Hypothesis testing
- Example: gasoline prices

Regression Analysis

Objective: Fit linear and nonlinear models to a bivariate data set.

- Predictors and responses
- Scatter plots
- Correlation and covariance
- Linear models
- Nonlinear models

Analysis of Variance

Objective: Consider the problem of determining differences in grouped data, including multiple comparisons between groups. Multiple groupings and multiple response variables are discussed.

- One-way ANOVA
- Two-way ANOVA
- N-way ANOVA
- Multivariate ANOVA

Multivariate Statistics

Objective: Extend the concepts discussed in previous sections to data sets with many variables. Specialized techniques for multivariate analysis and visualization are introduced.

- Multivariate plotting
- Principal component analysis
- Clustering