

# Software Carpentry Bootcamp using MATLAB

## Course Outline

### 1. The MATLAB Language and Desktop Environment

Objective: Import, organise and visualise data stored in multiple files.

- ▶ The MATLAB Desktop.
- ▶ Importing data:
  - from one file;
  - from multiple files.
- ▶ Vectors and matrices:
  - indexing;
  - concatenation;
  - removing missing values.
- ▶ Visualisation:
  - plotting;
  - annotation.
- ▶ Cells and structures.
- ▶ Saving data to MAT files.
- ▶ Scripts:
  - sections;
  - running;
  - publishing.

### 2. Algorithm Design in MATLAB

Objective: Develop and structure an algorithm to perform simple preprocessing, model-fitting and visualisation.

- ▶ Initial algorithm for 1D model-fitting:
  - formulating a linear regression model;
  - solving linear systems;
  - visualising the results.
- ▶ Generalising the algorithm to 2D model-fitting:
  - anonymous function handles;
  - surface plots.
- ▶ Code modularisation:
  - transferring code from scripts to functions;
  - local functions.
- ▶ Code robustness and flexibility:
  - parsing user-supplied input arguments;
  - defining flexible interfaces;
  - errors and error identifiers.

### 3. Test and Verification of MATLAB Code

Objective: Write function-based unit tests to formally test MATLAB algorithms.

- ▶ The MATLAB Unit Testing Framework:
  - overview;
  - function-based unit testing;
  - local functions.
- ▶ The test environment:
  - organising test data and test paths;
  - setup and teardown functions.
- ▶ Effective test design:
  - writing test functions;
  - testing robustness of functional interfaces;
  - testing numerical algorithms;
  - test design considerations.
- ▶ Running tests and evaluating the results.

### 4. Debugging and Improving Performance

Objectives: Use integrated MATLAB development tools to diagnose errors and identify potential for performance improvement. Write vectorised MATLAB code.

- ▶ Tools for Diagnosing Errors:
  - breakpoints;
  - directory reports.
- ▶ Tools for Measuring Performance:
  - timing functions;
  - the MATLAB Profiler.
- ▶ Improving Performance:
  - vectorisation strategies;
  - vectorising operations on cells and structures;
  - memory preallocation;
  - efficient memory management.