SCHOOL OF PHYSICS, ASTRONOMY & MATHEMATICS

4PAM1008 MATLAB

1 – Introduction to MATLAB

Dr Richard Greenaway

1 Introduction to MATLAB

1.1 What is MATLAB ?

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications. Features include:

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java™, COM, and Microsoft® Excel

The present course is designed to provide the *core* skills necessary to use MATLAB for manipulating, analysing and visualising data . It will not cover such things as generating user interfaces, advanced data plotting (handle-graphics) or any of the toolboxes except for the Symbolic Computation toolbox.

Figure 1 is a schematic of the MATLAB product family.

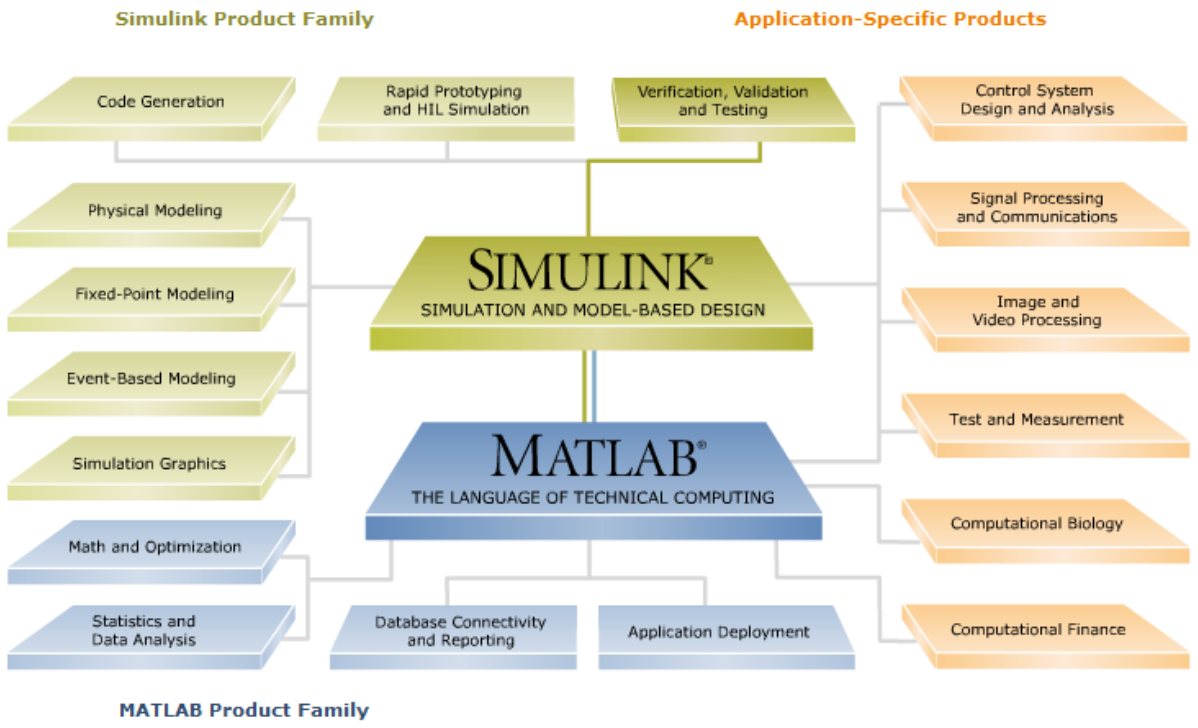


Fig. 1 Overview of MATLAB product family
(*Reproduced from MathWorks website*)

MATLAB comprises the core computing 'engine' and a set of *Toolboxes*, libraries of functions for specific applications such as statistics, image processing, symbolic computing etc. Simulink® is an optional add-on for simulation and model-based design for dynamic and embedded systems. **Simulink® will not be covered in this course.**

Why Use it?

According to MathWorks over a million people around the world use MATLAB for technical computing. It is widely used in industry and academia and according to InfoWorld (www.infoworld.com) it is a 'language on the rise'.

"Built for mathematicians to solve systems of linear equations, Matlab has found rising interest in the enterprise, thanks to the large volumes of data today's organizations need to analyze. Many of the more sophisticated statistical techniques that match people with advertisements, songs, or Web pages depend upon the power of algorithms like those solved by Matlab.

Expect Matlab use to grow as log files grow fatter. It's one thing for a human to look at the list of top pages viewed, but it takes a statistical powerhouse to squeeze ideas from a complex set of paths. Are people more likely to shop for clothes on Monday or Friday? Is there any correlation between product failures and the line that produced them?

MathWorks, the company behind Matlab, offers a diverse set of whitepapers showing how engineers are searching for statistical answers. Toyota Racing, for instance, plans its NASCAR entries by analyzing tests in wind tunnels and other labs. Canada's Institute for Biodiagnostics is searching for the best treatment for burns."

(InfoWorld, 25 October 2010)

1.2 Overview of the Interface

Fig. 2 shows the default layout for the MATLAB interface.

> ⚠ Be sure to use the default layout during the course so that your environment is consistent with the course materials.
>
> If your user interface does not look like that shown in Figure 2, you can select the default layout from the menu *Desktop* ►*Desktop Layout* ►*Default*.

The default interface shows four main windows.

- **The command window**
  The main window in the centre of the application. Here the user enters commands and expressions and output from calculations are displayed.

- **Current Folder window**
  This window shows the *working* folder. Commands to load and save data will target this folder by default.

- **Workspace Window**
  When data (matrices) are created they will be listed here.

- **Command History Window**
  Every command that the user enters in the command window will be logged here. From here the user can view their history and repeat commands previously entered by double-clicking or drag-and-drop.

> ⚠ The desktop version of MATLAB maintains a complete history from session to session. For some reason this is not the case with the networked version. Only commands entered in the current session will be available.

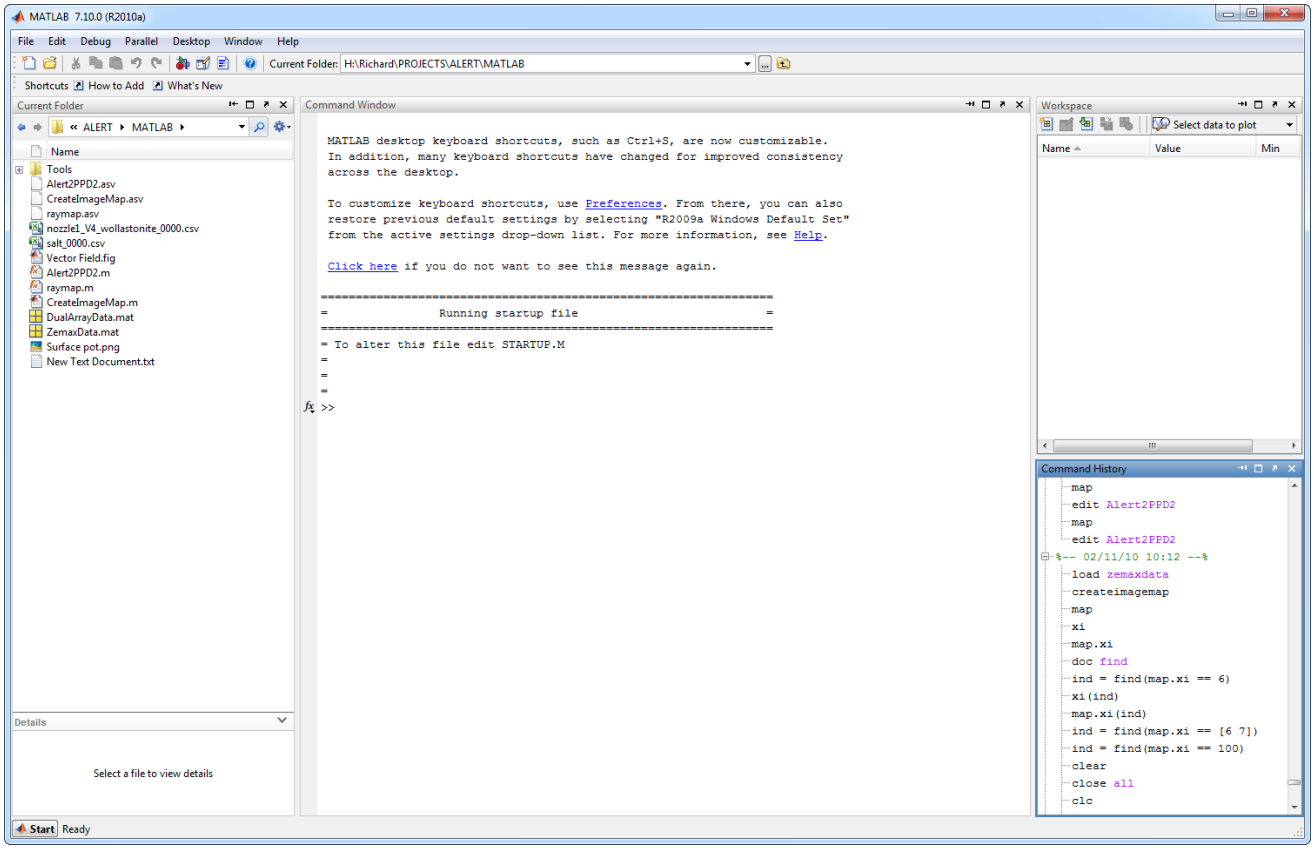


Fig. 2 MATLAB interface default layout

1.3 What Can MATLAB do?

1.3.1 MATLAB as a calculator

At its simplest; MATLAB can be used as a simple (albeit very expensive) calculator.

Exercise 1.1

We are going to do the following sum.

$$\frac{2+3^4}{11}$$

Type the following in the command window (don't include the command prompt '≫' which is there automatically).

≫ `(2 + 3^4)/11`

and press *enter*.

You should get the following output.

```
>> (2 + 3^4)/11

ans =

    7.5455
```

Whenever you type a command or expression, MATLAB will immediately provide output in the command window. The '+', '/' and '^' symbols are *operators* which perform arithmetic operations. Note the use of brackets to ensure the calculations are done in the corredct order.

Try the following and you will see that the answer is different.

```
» 2 + 3^4/11
```

1.3.2 MATLAB for Data Visualisation

MATLAB is a very powerful tool for quickly plotting and visualising data.

Exercise 1.2

Try this exercise. Don't worry about the details, simply type in the commands **exactly** as written.

```
>> z = peaks(30);
>> surf(z)
```

Remember that you must press **<enter>** before a command will execute.

If all goes well you should get the following 3-D plot

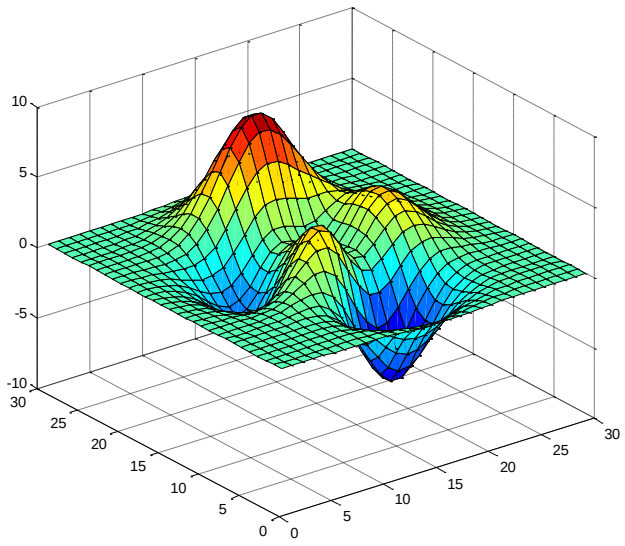


Fig. 3 Graph from Exercise 4

The function `peaks` returns a matrix of data which is a combination of Gaussian distributions. Its only purpose is for demonstrations. The function `surf` is one of the many plotting functions available in MATLAB and, as its name suggests, is used for plotting surfaces.

1.3.3 Data Analysis

Exercise 1.3

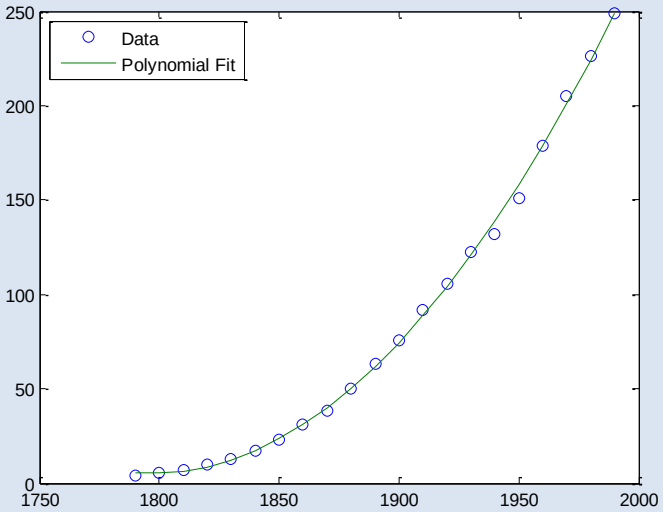
Here is a simple exercise in curve fitting. You will need to download the data from the Teaching Resources folder.

Data in MATLAB can be stored in **.mat** files and loaded using the `load` command. The example data used in this exercise comprises a list of dates (`cdate`) and the US population in millions (`pop`) corresponding to those dates. The function `polyfit` generates a polynomial (in this case a quadratic) to fit the supplied data and returns the polynomial coefficients. These are then passed to `polyval` to calculate equivalent population values using the polynomial model. We then plot the results,

Execute the following commands

```
>> load exercise_1_3;
>> p = polyfit(cdate,pop,2);
>> fitValues = polyval(p,cdate);
>> plot(cdate,pop,'o',cdate,fitValues,'-');
>> legend('Data','Polynomial Fit','location','NorthWest');
```

The result should be a plot like this.



1.3.4 Symbolic Computation

MATLAB can also be used to carry out symbolic computation (requiring a special add-on called a *toolbox*), that is it can be used to solve symbolic mathematical problems – differentiation, integration, simplification etc. Although primarily a *numerical* tool, the symbolic computation capibilities og MATLAB are very powerful and useful and these will be covered in the course.