Preface

Some programming languages have been used worldwide in every branch of scientific and technical activities from the sixties of the last century. MATLAB taking the inheritance of the former Fortran, Algol, Basic, Pascal, Ada and C, is now *The Language of Technical Computing* most known and used in academia, industries and services.

MATLAB is composed by a very large set of *functions*, properties of *graphics objects* and *operators*. Their list is in MATLAB Help browser both arranged alphabetically and subdivided into *categories* and, rewritten in a more suitable form, in our Appendixes 1, 2 and 3. Quite likely the reader browsing these large appendixes could not value the depths but certainly he can comprehend the great extent of MATLAB language.

Each of two official manuals (*MATLAB Programming* and *Using MATLAB Graphics*) has about 700 pages. The Help browser, an html book that is an integrant part of MATLAB, occupies probably more space. Then a very large textbook would be necessary to cover *all* the topics and it would too require an advanced scholarship of mathematics and a high computer and programming skill: it would be improbable to find someone interested to *all* subjects and to a similar book. If this has to be useful some choices must be made.

Firstly we assume our readers are keen people with, at least, a sound education of elementary mathematics (basic algebra and trigonometry) in a secondary school. They, as it is now usual, will be skillful with a PC and Internet too. The education of a higher school would be a fine edge but it is not indispensable.

Secondly, we identify and use some *functions*, *graphics objects* and *operators* as basic and others as applications or extensions of them. For

example we assume as basic a function that finds the square root of a number and as application one that determines the solutions of a complex equation. Besides we consider basic some graphics objects concerning simple 2-D plots and advanced those used, as example, for 3-D plots.

The *functions* we do not consider are however included in our long Appendix 2. We will discard, for example, functions concerning numerical calculus (vectors, matrices, systems of equations, integration, differential equations, *etc*.) or requiring specific competence (for example, object-oriented programming or handling hardware and software interfaces).

Neither will we add a chapter about Symbolic Math that, requiring an advanced mathematic expertise (for example, how to evaluate an indefinite integral), is beyond our postulated prerequisites.

Because visualization is the natural and mandatory complement of programming, even if a very complex and difficult task, we will deeply treat only some basic graphics objects (root, figure, axes, line, text and image), omitting all the others, and the large number of graphics functions necessary for programming a GUI, a Graphical User Interface.

It is an odd assumption a reader can both be interested and understand applications as how to illuminate a room, solve the heat equation, use the Fast Fourier Transform, estimate the water flow in a river, analyze a noisy signal, create a model of traffic flow, define the mortgage payments, develop a model for asset pricing or portfolio optimization, *etc*., or how to visualize a surface with the mesh or surf functions, plot volumetric data, make a real animation application, draw plant-like objects, create lighting effects to visualize fluid flow, *etc*.

Then we will not give *applications* requiring by the reader expertise in some specialized areas but only *examples*[1] (every accurately checked) devised for a deep insight of fundamental concepts of MATLAB language so that a user will be able to write clear and clean lines of code.

When the basic *functions*, *operators* and *properties* concerning *data* and *graphics objects* and the corresponding *exercises* will be well understood the reader, owning a specific competence, can apprehend using Help browser the remaining MATLAB features if and when they could be necessary to solve its own problems.

[1] In the website *www.antoniosiciliano.org* are allocated all the numbered examples present in the book and the colored figures that appear black and white here.

As example, our book *Optics Problems and Solutions*, containing a hundred of MATLAB programs, makes a large use of functions concerning numerical analysis.

Our final intent is to accustom the reader to a constant use of the MATLAB Help tools that could remain his sole tutor and, if this is not adequate to fulfil its need of assistance, to make him able to find solution with a patient but alert use of the "trial and error" method: trying out to run subsequently modified versions of the same program until the requested results are obtained. The structure of our book has been defined in order to achieve this purpose. Then with our selected topics of MATLAB as its basic foundations the reader will comprehend and master the programming tools necessary for applications in every fields of scientific, engineering, industrial and services.

We first introduce the Desktop, a powerful and friendly interface between the reader and the resources of MATLAB language. The standard procedures for writing a program are the arguments of the following chapters. A distinctive feature of MATLAB is the treatment of data as arrays of every type and number: the Chapter 2 gives a summary survey about data types and M-files that are deeply treated in subsequent chapters.

Chapter 3 treats formally the two basic manners to write a program. Even if common to all programming languages known from the first era of informatics, MATLAB treats them in a new and fine way, calling the main program a script and maintaining the old name *function* for a program that is executed only if called from a main program. MATLAB defines both as M-files when they are stored in a folder of a disk. Chapters 4 and 5 give the detailed formal treatment of all types of arrays (numerical, logical, literals) and the two special ones called Structures and Cells.

The Chapter 6 concerns the essential constituents of graphics objects. The argument is very complex: in this case the adjective means both "composed of many interconnected parts" and "intricate or complicated". The treatment is quite different from that present in the other MATLAB books we hope the reader will appreciate. We consider, and explain deeply conceptually and with examples, the main graphics objects, their properties and the functions that allow us to display the objects and control their properties. Two simple and preliminary cases are considered in Chapter 7: 2-D plots and images. The final Chapter 8 treats the various types of flow control that, exploiting the speed of a computer, are another classical feature of the programming languages from the first era.