Getting Started in MATLAB

by Greg Fasshauer

If you're having trouble getting started with MATLAB you should read through these three pages, go through the *Very Elementary MATLAB Tutorial* provided by The MathWorks at http://www.mathworks.com/academia/student_center/tutorials/intropage.html, or read the first chapter or two of the *Getting Started Guide* mentioned below.

Another option is to use the information provided with MATLAB itself. Unless it has been turned off – an info bar appears at the top of MATLAB's command window. You can click on any one of the three links (*Video*, *Demos*, or *Getting Started*). Each of these should start up a file that was part of the MATLAB installation. Depending on your machine you will find *Video* at some place like C:/Program Files/MATLAB/R2009a/toolbox/matlab/demos/html/GettingStartedwithMATLAB.html and it should start up in your web browser. The link *Demos* will open up the MATLAB Help browser with all sorts of demos, and the *Getting Started* link operates similarly. The corresponding PDF document is also available on the web at http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/getstart.pdf.

Additional links to even more information are provided on my webpage http://math.iit.edu/∼fass/-100.html.

1 What is MATLAB?

MATLAB is widely used in many areas of applied mathematics and engineering. MATLAB stands for MATrix LABoratory and the software uses vectors and matrices as basic building blocks. This makes the software particularly useful for dealing with linear algebra problems. However, it also means that we have to learn to think "the MATLAB way" if we want to take full advantage of these building blocks to create fast and efficient code. Of course MATLAB can also be used to solve many other problems such as (nonlinear) algebraic and differential equations, or for the numerical evaluation of integrals (quadrature). In addition to its computational engine MATLAB provides a powerful graphical interface that allows us to produce both 2D and 3D plots. While much can already be accomplished in an interactive mode, MATLAB is also one of the easiest programming languages for solving mathematical problems since it provides much higher-level access to mathematical quantities than traditional programming languages such as, e.g., C, C++, Fortran or Java. Finally, while we will not be discussing any of these, MATLAB's basic capabilities can be extended by calling functions defined in additional *toolboxes*.

2 How to Start and Exit MATLAB

For all purposes I can imagine you will want to use MATLAB via its standard windows-based interface.

All IIT computer labs should have MATLAB installed and an icon should be visible on the Desktop. Just double-click this icon to start the program. (This procedure should work similarly on other platforms such as Macs or Unix (Linux) machines.)

The most important parts of MATLAB for you will be the Command window and the Editor (see below). In the Command window you communicate with MATLAB. For example, when you want to run a program you have written for MATLAB (using a so-called M-file) you start that program in the command window by typing its name at the `>>` prompt. Usually you will write your program code in a separate window (the MATLAB Editor window), save it to your hard drive, and then run it from the Command window. While the MATLAB Editor is probably the most natural editor to use, any editor can be used as long as you save your program as a text file with a `.m` extension.

If you're using the standard windows interface to MATLAB then you are able to get help in the separate Help window. You can also just type `help` (or `help <command name>`) at the command prompt.

In an emergency (such as a run-away loop) you can interrupt MATLAB by typing Ctrl-C (note that sometimes it may take MATLAB a while to "come back" from heavy calculations).

Once you have finished your work you can exit MATLAB by either typing `quit` at the prompt (`>>`) in the Command window, by going to the File→Exit menu, or by "x-ing out of" the Command window in the usual way.

In addition to the windows-based interface with all its bells and whistles MATLAB also has a command-line interface that can be invoked by using additional switches such as `matlab -nodesktop`. However, you will probably not find much use for this interface.

3 Using the MATLAB Editor

While you can enter individual MATLAB commands interactively one at a time in the Command window, you will often want to combine a sequence of commands into a program (also called a script file or function file). You need to write such programs in a separate editor (see above). If the Editor does not open by itself when you start MATLAB you can invoke it via the File→New→M-File menu (for a new file) or File→Open menu (for an existing file). Basic use of the editor is straightforward. However, many advanced features such as adding breakpoints to your code for debugging purposes are also possible.

While typing your code in the editor, no commands will be performed! In order to run a program do the following:

- In the Editor save your code as an M-file with some filename you pick. (MATLAB should automatically add the `.m` extension that is required for the file to be recognized as a MATLAB program file).
- Go to the Command window. Make sure the folder your Command window is looking at is the same one you saved your program in! (If you need to change folders you can do this either with the `cd` command at the command prompt, or by using the Current Directory pane that you can add to your MATLAB window using the Desktop menu.)
- Run the program by entering its name (without the `.m` extension) at the command prompt. If your code contained an error, MATLAB will interrupt execution of the program and provide you with an error message. You can click on the error message, and will be taken to the corresponding place in the code in the Editor (make sure you click on the error message that corresponds to "your" code, not some built-in MATLAB function).

An alternative way to get a set of commands which you may have tried out interactively in the Command window into a script file in your Editor so that you can save and use them later is to highlight the commands in the Command History window, right-click your mouse and choose the *Create M-file* option. This will automatically open the MATLAB Editor window with a new M-file containing the selected commands. You can now proceed as above.

In order to save both your commands and their output while working in the Command window you can save everything to a so-called diary file (a simple text file that you can view with any text editor, such as the MATLAB Editor). To this end you would type `diary filename` at the command prompt to begin the recording process and `diary off` to end it. However, only the output that is displayed in the Command window is stored in the file named `filename`, graphical output does not get preserved in the diary. Typing `help diary` provides more details.

4 Further Details

Much more information about programming in MATLAB along with examples is provided in the following documents (all available through the class webpage http://math.iit.edu/∼fass/100.html):

- MATLAB *7: Getting Started Guide*, which should also be available directly through your MATLAB installation,
- *Practical Introduction to* MATLAB *(kind of outdated, for* MATLAB *Version 5)* by Mark Gockenbach,
- Chapter 1 of *Numerical Computing with* MATLAB by Cleve Moler,
- the sample scripts discussed in class should also be helpful.

The following books are also recommended:

- *Experiments with* MATLAB by Cleve Moler, the textbook for MATH 100 which can be downloaded for free at http://www.mathworks.com/moler/exm/chapters.html.
- *Learning* MATLAB by Tobin A. Driscoll, see info at http://www.ec-securehost.com/SIAM/OT115.html. The book costs $19.60 for SIAM members ($28.00 otherwise).
- *Numerical Computing with* MATLAB by Cleve Moler. The book can either be downloaded for free from http://www.mathworks.com/moler/chapters.html, or purchased here http://ec-securehost.com/SIAM/ot87.html for $34.65 (SIAM member price) or $49.50 (others). We currently use this as a textbook for MATH 350.
- MATLAB *Guide* (2nd ed.) by Desmond J. Higham and Nicholas J. Higham, see info at http://www.ec-securehost.com/SIAM/ot92.html. A much more comprehensive book about MATLAB. $35.00 for SIAM members ($50.00 otherwise).

Please note that **every IIT applied math student can become a SIAM member for free** (see http://www.siam.org/students/memberships.php. You may also contact siam@math.iit.edu for more information.