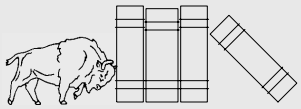
MATLAB for Windows



JumpStart
Computing and Information Technology

Introduction

MATLAB (short for "MATrix LABoratory") is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment. *MATLAB*'s strength is the ease with which it manipulates rectangular arrays of numbers (matrices), relieving the user of explicit dimensioning details.

Start and Quit *MATLAB*

To start *MATLAB*, click on **Start** button, then select **Programs**, **Matlab**, and then **MATLAB**. A new window opens. *MATLAB*'s command line prompt is a pair of greater-than symbols, >>.

To end your *MATLAB* session, type ***exit*** or ***quit*** or choose **Exit MATLAB** from the **File** menu.

File Conventions

MATLAB expects certain file extensions when some commands are executed. Here is a list of file extensions:

File Contents	*Extension*
MATLAB script file	*.m*
MATLAB binary file	*.mat*

MATLAB reads and executes commands from standard input, as well as from ASCII script files with *.m* extension (called M-files, which can be created with MATLAB built-in editor or *Notebook*). To execute the commands residing in a file named *myproject.m*, simply type the file name without the *.m* extension at a *MATLAB* prompt:

```
myproject
```

MATLAB executes each command from *myproject* until complete, then returning you to a prompt. Comments can be used to document your work; *MATLAB* treats any text following a % sign as a comment.

Help in MATLAB

MATLAB provides an effective online help facility, available by typing ***help***. *MATLAB* responds by displaying a list of help topics, along with a brief description of each topic. You can get specific help by typing ***help*** with a topic name. For example, typing ***help graph3d*** lists the commands that relate to three dimensional graphics; typing ***help surf*** provides detailed help on *MATLAB*'s surface plotting command ***surf***.

Online help is also available from the menu bar located at the top of the *MATLAB* window.

To get a quick introduction to *MATLAB*'s features, run the demo program by typing ***demo***.

MATLAB Command Syntax

MATLAB is case sensitive. All built-in commands are in lower case. A command normally terminates with a carriage return. Including a semicolon (;) at the end of a command suppresses *MATLAB*'s echoing to the terminal. Typing ***b = 2*a***; stores the result of ***2*a*** in variable ***b***, but does not display the result (This is useful when dealing with large sets of numbers).

Matrices

MATLAB works with essentially one kind of object: a matrix of numbers (which could include complex elements). Scalars are 1-by-1 matrices and vectors are 1-by-n or n-by-1 matrices.

When entering a matrix, separate columns by space or commas and rows by semicolons. For example, typing

```
A = [1 2; 3 4]
```

results in

```
A =
    1     2
    3     4
```

MATLAB stores the above 2-by-2 matrix into the variable *A* for later use. To retrieve a variable, simply type its name (e.g., ***A***).

Matrix and Array Operations

Matrix operations are fundamental to MATLAB and are based on principles of linear algebra. Matrix multiplication, division, power and transpose are denoted by symbols ***, /, ^** and **'**.

Array operations refer to element-by-element arithmetic operations, rather than the usual linear algebraic operations. Array operations are denoted by preceding an operator with a period (**.**) such as ***.****, **./** and **.^**. For addition and subtraction, array and matrix operations are same.

See the *Examples* section for a comparison of matrix and array operations.

The Colon (:) Notation

The colon, **:**, is an important notation. It can be used to generate vectors and access submatrices. For example

```
x=0 : pi/4 : pi
```

results in

```
x=
  0 0.7854 1.5708 2.3562 3.1416
```

Combination of vectors can generate matrices. For example

```
y=exp(-x).*sin(x);
z=[x;y]
```

Document Number: JST-WIN-006 Revised: August 1998
Published by: Academic Services, Computing and Information Technology
State University of New York at Buffalo
JumpStart is a series to help users get started with language compilers, application and utilities with minimal startup time.

For additional help contact the CIT Help Desk, 216 Computing Center, at (716) 645-3542 or send email to *cit-helpdesk@buffalo.edu.*

produces:

```
z=
   0 0.7854 1.5708 2.3562 3.1416
   0 0.3224 0.2079 0.0670 0.0000
```

The colon notation allows you to access any portion of a matrix. For instance

```
z(:,2:5)
```

specifies the 2-by-4 submatrix that consists of the last four elements in all rows. Please note that using the colon alone denotes all of a corresponding row or column.

Editor/Debugger

MATLAB's built-in Editor/Bebugger can be used to edit and debug command/script files. To access it type

```
edit
```

Saving your work

Choose **Save Workspace As...** from the **File** menu to save your work into a *.mat* file. Files with a *.mat* extension are in binary format and are not human readable. To open an existing *.mat*, select ***load Workspace*** from the **File** menu.

Printing

Select **Print** from the **File** menu to print a copy of your *MATLAB* session.

Examples

MATLAB's command language is expressive, extensible, and easy to use; using *MATLAB* interactively by entering commands at its >> prompt is an effective way to learn.

The following commands help illustrate some of *MATLAB*'s features:

Example	*Description*
clear	Clear all variables from memory.
a = [1 2; 3 4]	Create a simple 2x2 matrix.
b = inv(a)	Invert matrix ***a*** and store it as variable ***b***.
c = a * b	Matrix multiplication of ***a*** and ***b***.
c = a .* b	Array operation (note the decimal point in the operator!).
c(:, 2)	The 2nd column of matrix ***c***.
c(2, :)	The 2nd row of matrix ***c***.
who	List the active variables now.
x = 0: pi/30 : 4*pi	Create a row vector whose elements range from 0 to 4 pi, in increments of pi/30.
x = x'	Take the transpose of ***x***, and store the resulting column vector as ***x***.
y = exp (-0.3 * x) .* sin (x)	Create column vector ***y***. as a function of column vector ***x*** (note the ***.**** operator!).
plot(x,y)	Plot the column vectors ***x*** and ***y***.
title('Decaying Sinusoid')	Create a plot title.
xlabel('Time (sec)')	Label the x-axis.
ylabel('Positio n (in)')	Label the y-axis.
print -dps decay	Create a PostScript plot file *decay.ps*.
clf	Clear the displayed figure.
[x, y] = meshgrid(-1: 0.05 :1 , -1: 0.1 : 1)	Generate matrices x and y to support 3-D surface plotting over the provided intervals.
z = sin(5 * x .* y);	Create matrix z, from x and y.
surf(x,y,z)	Plot the surface z = sin(5xy).
view([20 60])	Change the view and replot (azimuth & elevation).

Documentation

For a complete description of *MATLAB*, refer to

MATLAB User's Guide

published by The MathWorks, Inc.

To access online manuals, click on **Help** and then **Help Desk**.