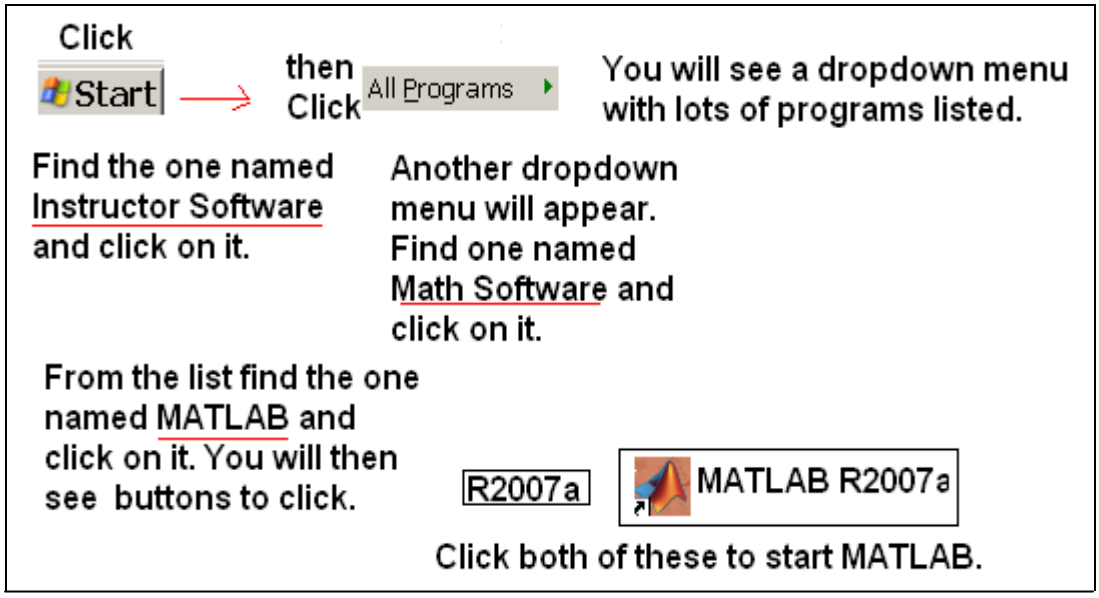
What is Matlab?

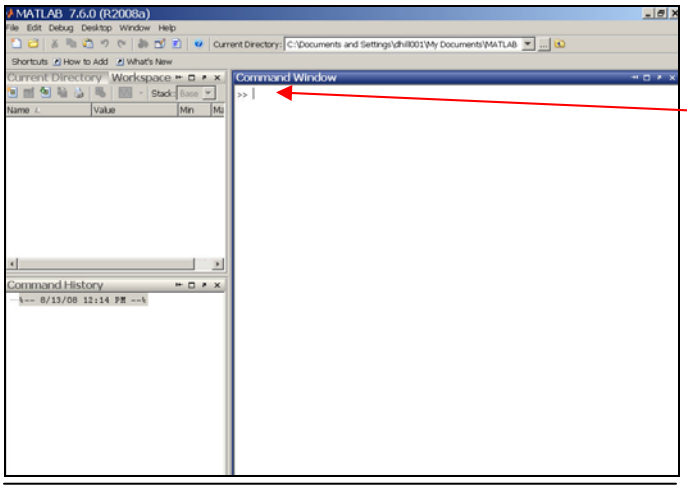
Matlab (short for MATrix LABoratory) is a language for technical computing, developed by The Mathworks, Inc. (A matrix is a rectangular array or table of *usually* numerical values.) Matlab provides a single platform for computation, visualization, programming and software development. All problems and solutions in Matlab are in a notation very similar to the way mathematics is written. In addition, you can use Matlab to build Graphical User Interfaces (GUIs) so that you can develop user-friendly custom software.

Matlab is widely used in all areas of applied mathematics in education and research at universities and in industry. Matlab is a great tool for solving a wide variety of problems in science, engineering, and finance. In addition, Matlab has powerful graphic tools and can produce nice pictures in both 2D and 3D. It is also a programming language (similar to C) and is one of the easiest programming languages for writing mathematical programs.

Starting Matlab Logon at MSRC using desktop icon  . To logon onto a Temple workstation at the Tech Center, follow the directions below.

You will see a screen like the following:





This is where you enter commands. See the next paragraph.

The Matlab environment

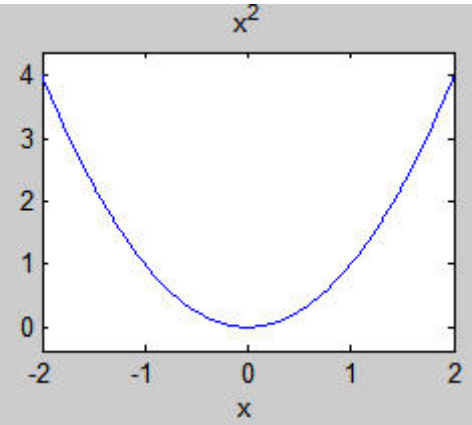
The Matlab environment consists of menus, buttons and a writing area similar to an ordinary word processor. There are plenty of help functions that you are encouraged to use. The writing area that you will see when you start Matlab, is called the ***command window***. In this window you give the commands to Matlab. For example, when you want to run a program you have written for Matlab you start the program in the command window by typing its name after the prompt. (**Matlab's prompt is >>.**) Once you have typed the command you wish Matlab to perform, press Enter. The command window is also useful if you just want to use Matlab as a scientific calculator or as a graphing tool. If you write longer programs, you will find it more convenient to write the program code in a separate window, and then run it in the command window.

Example: One way to plot the parabola $y = x^2$ over interval [-2, 2] in Matlab is to use the following command.

```
>> ezplot('x^2',[-2,2])
```

The expression for the formula is included between quotes. The command used here is **ezplot**, which we read as "easy plot".

The result is shown in a **graphics window** which is shown below.



Quitting Matlab

To quit Matlab, at a Matlab prompt type command **quit** or **exit**, then press Enter.

Using Matlab as a (scientific) calculator

Matlab's computational engine uses (about) 15 decimal digits in computations (unless directed to do otherwise). So it can be used as a powerful calculator for arithmetic expressions and the calculation of values of function expressions.

Arithmetic Operations	Matlab command
+ (addition)	+ a + b
- (subtraction)	- a - b
x (multiplication)	* a*b (a times b)
/ (division)	/ a/b, b ≠ 0
^ (exponentiation)	^ a^b (a raised to the b power a^b)

WARNING: You must indicate multiplication using the * .

Examples: In expressions it good practice to use parentheses as illustrated below. This is particularly important for fractions which are used in Matlab.

Matlab expressions	Value shown in format long e	Value in format short
(1.5 - 4.67)*(9/3.1)	-9.203225806451613e+000	-9.2032
(5.2 - 3.72)^(1/2)	1.216552506059644e+000	1.2166
(7.02 + 12.3)/(4.25 – 3.125)	1.717333333333333e+001	17.1733

In Matlab you can control how a numerical value is displayed by using what are called format commands. After a Matlab prompt type command **format long e** or **format short** to get the displays above. Other such commands are **format long** and **format short e**.

The number π (pi) in Matlab can be accessed by using the name **pi**. Here is what is displayed for π in various formats.

Format short display for **pi**	3.1416
Format long display for **pi**	3.141592653589793
Format short e display for **pi**	3.1416e+000
Format long e display for **pi**	3.141592653589793e+000

Note that the "e formats" display in scientific notation; 5.6782e+002 means 5.6782×10^2.

Also note that the last decimal digit in a "short display" is obtained by rounding.

Regardless of the display format used to show values on the screen computations are done with 15 digit arithmetic.

Use command **help format** for the description of other display formats.

Special case: A value which is exactly zero will be displayed as a single zero. If you see 0.0000, the value is not exactly zero. Change to a long format and re-display it.

Expressions for functions in Matlab: Illustrations of ALGEBRAIC Expressions and their MATLAB form.

Standard Algebra Form	MATLAB Representation
$3x^2 - 5x + 1$	3*x^2-5*x+1 multiplication must be indicated using * and exponents require an ^
$\frac{2x-3}{4-7x}$	(2*x-3)/(4-7*x) the numerator and denominator must be enclosed in parentheses if there is more than 1 term
$\sqrt{x^2+1}$	sqrt(x^2+1) or (x^2+1)^.5 or (x^2+1)^(1/2) using sqrt is preferred; fractional exponents must be enclosed in parentheses
$4x^3\sqrt{3x+7}$	4*x^3*sqrt(3*x+7) note the use of * for multiplication, ^ for exponentation and sqrt for the square root

Matlab has many other functions including logarithms, trigonometric, absolute value, and the exponential function.

Plotting 2 dimensional graphs

The ezplot command

The easiest way to use ezplot is to express a function is between quotes:

Example: **ezplot('x^2 - 2*x + 1')**

The **default domain** -2*pi < x < 2*pi.

To change the domain use

EZPLOT(FUN,[A,B]) which plots FUN(x) over $A \leq x \leq B$.

Example: **ezplot('x^2 - 2*x + 1', [-1, 4])** or **ezplot('x^2 - 2*x + 1', -1, 4)**

Explicit commands to generate a plot

Instead of using **ezplot** we can construct a set of points in the domain, evaluate the function at these points, and then plot the ordered pairs. This requires more steps, but you have more control over the image and labels along the graph. For the image you can control color and line type. You can also reset the axes.

Example: Sketch a graph of y = f(x) = $\sqrt{x^2+1}$ over [-1, 2].

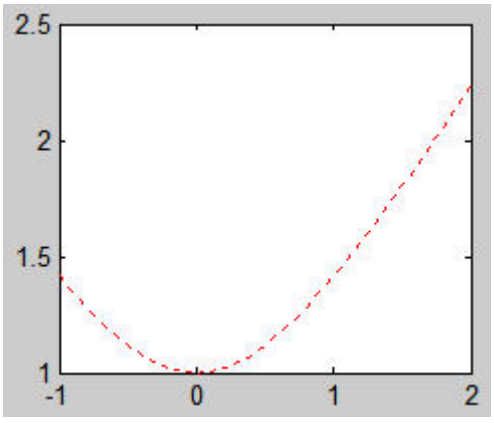
f = 'sqrt(x ^ 2 + 1)' ; ← gives the function a name; note using the quotes. The semicolon at the end suppresses the display of the contents of variable f.

vf = vectorize(f); ← using command **vectorize** lets us evaluate the function at all the domain points that we will define below at one time and thus avoid using a loop

x = -1:0.5:2; ← here we define the domain starting at -1 in steps of 0.5 until 2; we need the set of domain points to be the same as the variable in the definition of the function f.

y = eval(f); ← this evaluates the function f at the values of x

plot(x, y, ' : r') ← this generates the graph using red dots



The following commands are useful when plotting:

Graphing functions	Matlab command
Label the horizontal axis.	**xlabel('text')**
Label the vertical axis.	**ylabel('text')**
Attach a title to the plot.	**title('text')**
"Keep plotting in the same window."	**hold on**
Turn off the "keep-plotting-in-the-same-window-command".	**hold off**

Note that all text must be put within single quotes like ' '.

Various line types, plot symbols and colors may be obtained with command **plot(x,y,S)** where S is a character string, that is, uses single quotes to enclose one or two elements from any or all the following 3 columns. For example **plot(x,y,'r-')** plots the ordered pairs and connects them with a solid red curve.

Colors		Point Symbols		Line Forms	
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		