



MATLAB

1. THE BASICS.



Accelerating the pace of engineering and science



United States ▶ | Contact Us | Store



Create Account | Log In

Products & Services Solutions Academia Support User Community Events Company

Products & Services > MATLAB

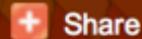
MATLAB

MAJOR UPDATE

The Language of Technical Computing

```
0.8147    0.0975    0.1576    %s = fftshift(fft2(1));
0.9058    0.2785    0.9716    colormap hot;
0.1270    0.5469    0.9572    set(gca,'Position');
0.9134    0.9575    0.4855    %pos = (ceil((n-1)/4)*ax(3)+ax(1));
0.6324    0.9649    0.8003    pos = ((2-floor((n-1)/4))/3)*ax(1)+ax(2);

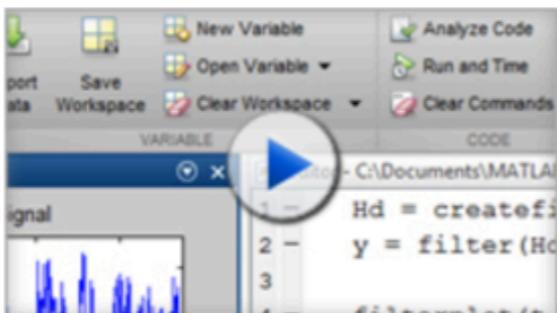
```



Overview

Videos & Examples

Webinars



Product Overview 2:04



Documentation



Functions



Data Sheet



Key Features

MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java™.



TRY OR BUY

Contact Sales
Product Trial
Pricing and Licensing

You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

R2012b

See the New MATLAB Desktop



Getting Started with MATLAB

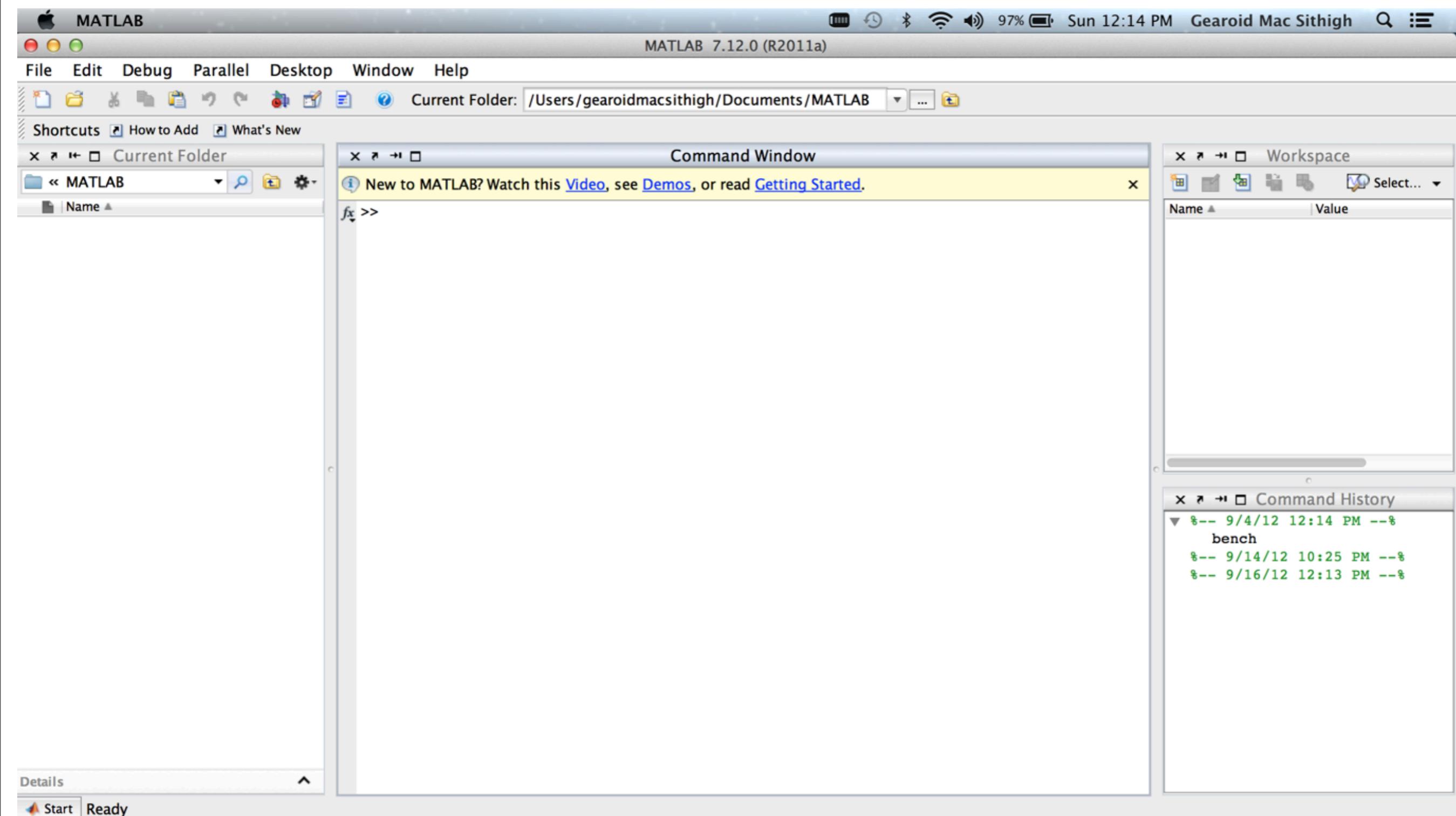


MATLAB and Simulink
Student Version

WHERE CAN I ACCESS MATLAB?

- 1. MATLAB IS INSTALLED ON ALL THE CLC MACHINES.**
- 2. STUDENT EDITION OF MATLAB MAY BE PURCHASED FOR \$99 FROM THE MATHWORKS WEB-SITE.**
- 3. THE IT WALK-IN CENTER WILL INSTALL MATLAB ON A PERSONAL LAP-TOP-WINDOWS / MAC WITH LION OR MOUNTAIN LION - FOR \$25.**

COMMAND WINDOW



ELEMENTARY MATH FUNCTIONS

$\sin(x)$	$\sin(x)$
$\cos(x)$	$\cos(x)$
$\tan(x)$	$\tan(x)$
$\text{asin}(x)$	$\arcsin(x)$
$\text{acos}(x)$	$\arccos(x)$
$\text{atan}(x)$	$\arctan(x)$
$\text{atan2}(x,y)$	Four quadrant arctan
$\exp(x)$	e^x
$\log(x)$	$\ln(x)$
$\log_{10}(x)$	$\log_{10}(x)$
$\text{floor}(x)$	Largest integer $\leq x$
$\text{sqrt}(x)$	\sqrt{x}
$\text{abs}(x)$	$ x $

BASIC MATLAB COMMANDS

format

```
>> format short e;  
>> z=pi;  
>> z  
  
z =  
  
3.1416e+00  
  
>> format short;  
>> z1=pi;  
>> z1  
  
z1 =  
  
3.1416
```

NOTICE THAT ; SUPPRESSES OUTPUT.

```
>> format long;  
>> z2=20*pi;  
>> z2  
  
z2 =  
  
62.831853071795862  
  
>> format long e;  
>> z3=150*pi;  
>> z3  
  
z3 =  
  
4.712388980384690e+02
```

MATLAB FUNCTIONS

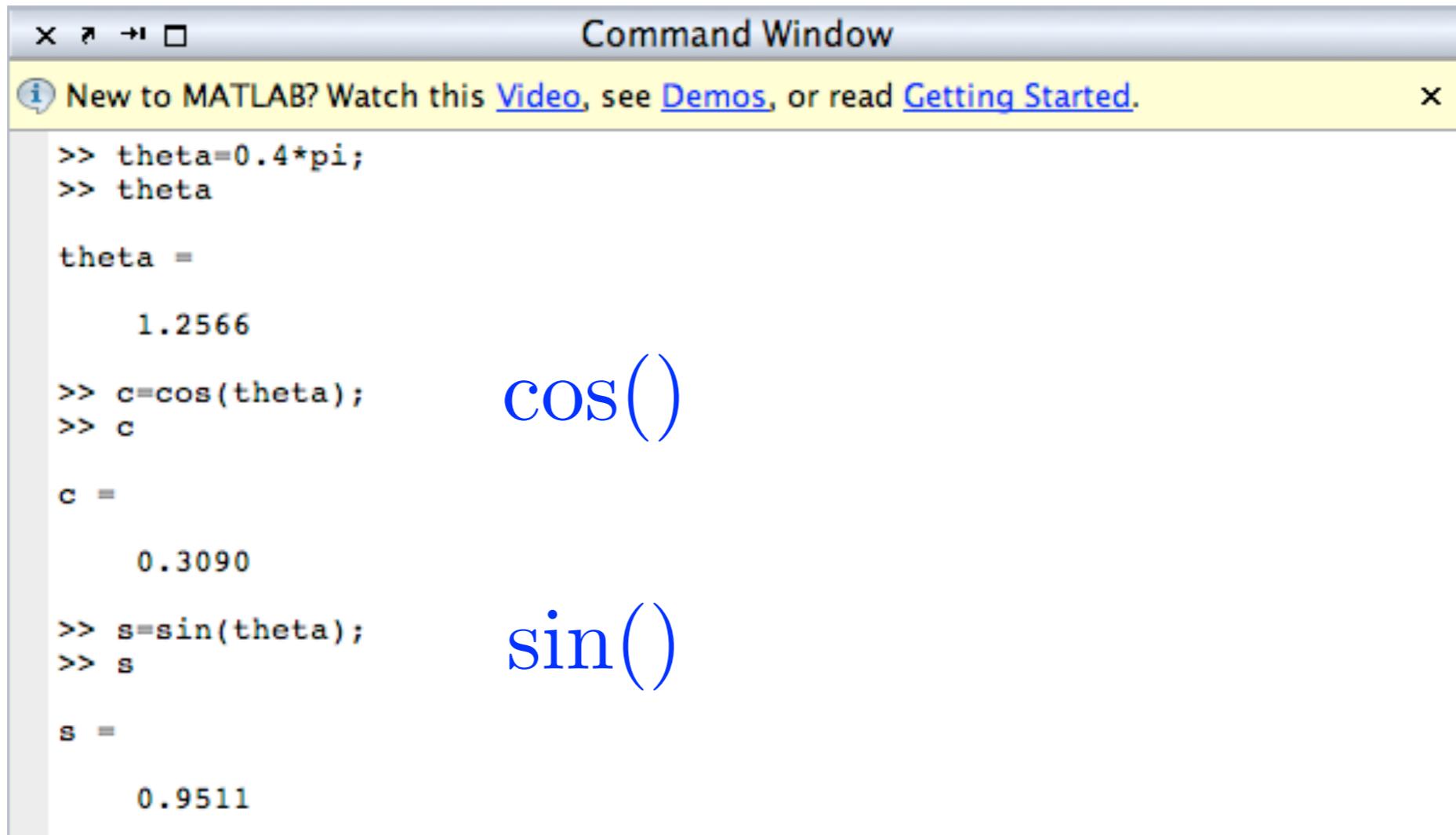
```
>> H=3^4;  
>> H  
  
H =  
  
81
```

power

```
>> G=sqrt(25);  
>> G  
  
G =  
  
5
```

square root

MATLAB FUNCTIONS



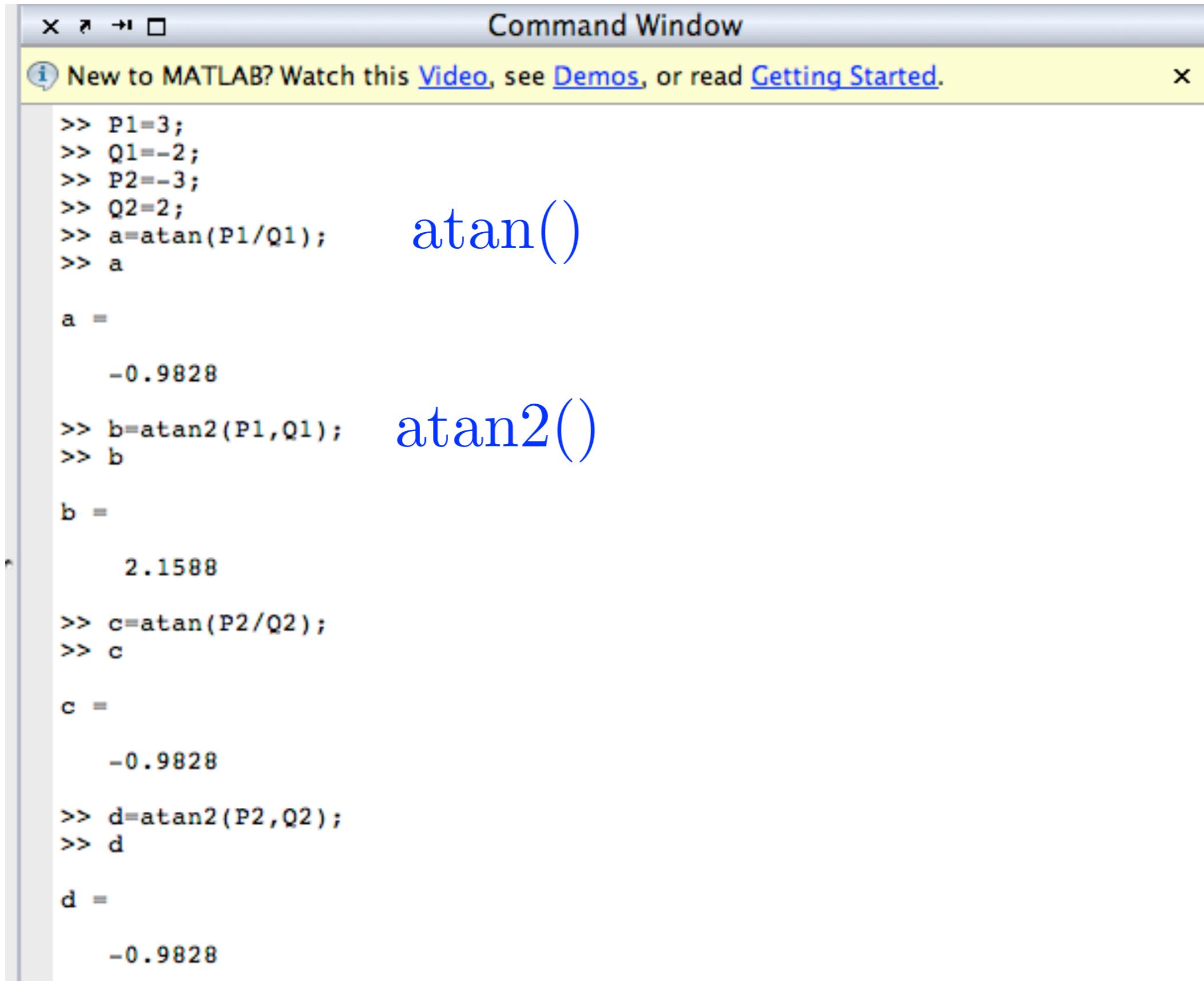
The image shows a screenshot of the MATLAB Command Window. The window title is "Command Window". A yellow status bar at the top says, "New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)". The command history is as follows:

```
>> theta=0.4*pi;
>> theta
theta =
    1.2566
>> c=cos(theta);
>> c
c =
    0.3090
>> s=sin(theta);
>> s
s =
    0.9511
```

Two large blue text labels are overlaid on the right side of the window: "cos()" above the sine calculation and "sin()" below it.

ARGUMENTS OF TRIG. FUNCTIONS IN RADIANS!

MATLAB FUNCTIONS



The image shows a screenshot of the MATLAB Command Window. The window title is "Command Window". A yellow status bar at the top contains the text: "New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)". Below this, the command window displays the following MATLAB code and its output:

```
>> P1=3;
>> Q1=-2;
>> P2=-3;
>> Q2=2;
>> a=atan(P1/Q1);      atan()
>> a
a =
-0.9828

>> b=atan2(P1,Q1);    atan2()
>> b
b =
2.1588

>> c=atan(P2/Q2);
>> c
c =
-0.9828

>> d=atan2(P2,Q2);
>> d
d =
-0.9828
```

MATLAB FUNCTIONS

```
>> a=10*pi;  
>> D=floor(a);  
>> D
```

D =

31

```
>> c=-10*pi;  
>> H=floor(c);  
>> H
```

H =

-32

>>

floor

floor(b)=LARGEST INTEGER NO GREATER THAN b

BASIC MATLAB COMMANDS

ENTERING MATRICES

```
>> K=zeros(3,3);  
>> K
```

K =

0	0	0
0	0	0
0	0	0

```
>> L(1,1)=10;  
>> L(1,2)=-4;  
>> L(1,3)=1;  
>> L(2,1)=0;  
>> L(2,2)=8;  
>> L(2,3)=-2;  
>> L(3,1)=4;  
>> L(3,2)=0;  
>> L(3,3)=5;  
>> L
```

L =

10	-4	1
0	8	-2
4	0	5

```
>> J=[4 0 2;1 5 2;0 2 5];  
>> J
```

J =

4	0	2
1	5	2
0	2	5

BASIC MATLAB COMMANDS

whos

```
>> whos
  Name      Size            Bytes  Class       Attributes
  J          3x3              72  double
  K          3x3              72  double
  L          3x3              72  double
```

TELLS YOU WHAT'S
CURRENTLY IN THE
WORKSPACE

MATRIX OPERATIONS

```
>> A=[2 3;1 5];
>> A
A =
    2     3
    1     5

>> B=2*A;
>> B
B =
    4     6
    2    10

>> C=A+B;
>> C
C =
    6     9
    3    15

>> D=3*A-2*B;
>> D
D =
   -2    -3
   -1    -5
```

MULTIPLICATION BY A SCALAR

MATRIX ADDITION

LINEAR COMBINATION

MATRIX OPERATIONS

Command Window

(i) New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> A=[1 2;0 4];
>> B=[5 0;1 3];
>> A
A =
    1     2
    0     4
>> B
B =
    5     0
    1     3
>> C=A*B;
>> C
C =
    7     6
    4    12
>> D=B*A;
>> D
D =
    5    10
    1    14
```

MATRIX
MULTIPLICATION

BASIC MATLAB COMMANDS

MATRIX OPERATIONS

```
>> Q=[5 0 0;0 4 0;0 0 8];  
>> H=inv(Q);  
>> H
```

H =

```
2.00000000000000e-01      0      0  
0      2.50000000000000e-01      0  
0          0      1.25000000000000e-01
```

```
>> R=[5 0 0;0 0 0;0 0 8];  
>> M=inv(R);  
Warning: Matrix is singular to working precision.  
>> M
```

M =

```
Inf  Inf  Inf  
Inf  Inf  Inf  
Inf  Inf  Inf
```

MATRIX INVERSION

MATRIX OPERATIONS

```
>> A=[5 0 2;1 4 1;2 2 5];
>> A
A =
      5     0     2
      1     4     1
      2     2     5

>> B=[5;7;2];
>> B
B =
      5
      7
      2

>> C=A\B;
>> C
C =
    1.3077
    1.6154
   -0.7692

>> D=B\A;
>> D
D =
    0.4615    0.4103    0.3462
```

LEFT DIVISION

IF A IS A SQUARE MATRIX,
 $A \setminus B = \text{inv}(A) * B.$

IN GENERAL, THE COMMAND
 $A \setminus B$ TELLS MATLAB TO FIND X
SUCH THAT $A * X = B$, PROVIDED
THAT EQUATION MAKES
DIMENSIONAL SENSE.

OPTIMAL BOGUS SOLUTION !!

f_x >> |

MATLAB FUNCTIONS

size, max

```
>> q=[0:.5:20];
>> size(q)

ans =

1      41

>> max(size(q))

ans =

41
```

RELATIONAL OPERATORS

MATLAB	MATH
<	<
>	>
\leq	\leq
\geq	\geq
$==$	$=$
\neq	\neq

LOOP STATEMENTS

for..... end	LOOP STATEMENT
if.....else..... end	CONDITIONAL STATEMENT
if...else if ... else ... end	COMPOUND CONDITIONAL STATEMENT
if break ...end	BREAKS LOOP WHEN CONDITION IS MET

MATLAB PROGRAMMING

'for' STATEMENT

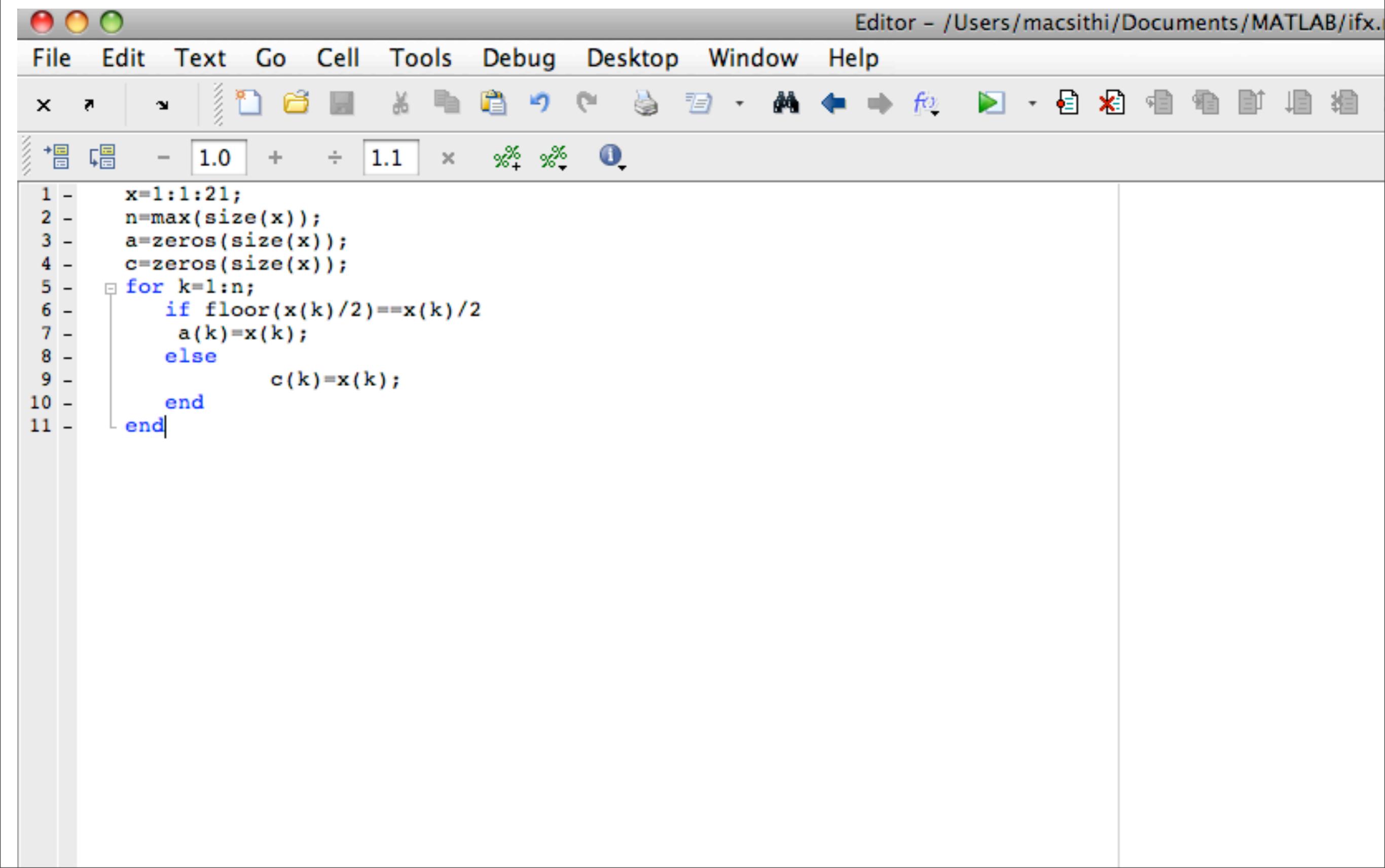
The screenshot shows a MATLAB IDE interface. The top menu bar includes Preview, File, Edit, View, Go, Tools, Bookmarks, Window, and Help. The title bar indicates the file is 'Editor - /Users/macsthithi/Documents/MATLAB/forx.m*'. The main workspace displays the following MATLAB script:

```
1 - a=0:.02:5;
2 - n=max(size(a));
3 - b=zeros(size(a));
4 - for k=1:n;
5 -     b(k)=3*a(k)-5;
6 - end
```

The code defines a vector 'a' from 0 to 5 with a step of 0.02, determines its size, creates a zero matrix 'b' of the same size, and then iterates through each element 'k' from 1 to n, calculating b(k) = 3*a(k) - 5. The 'for' statement is highlighted in blue.

MATLAB PROGRAMMING

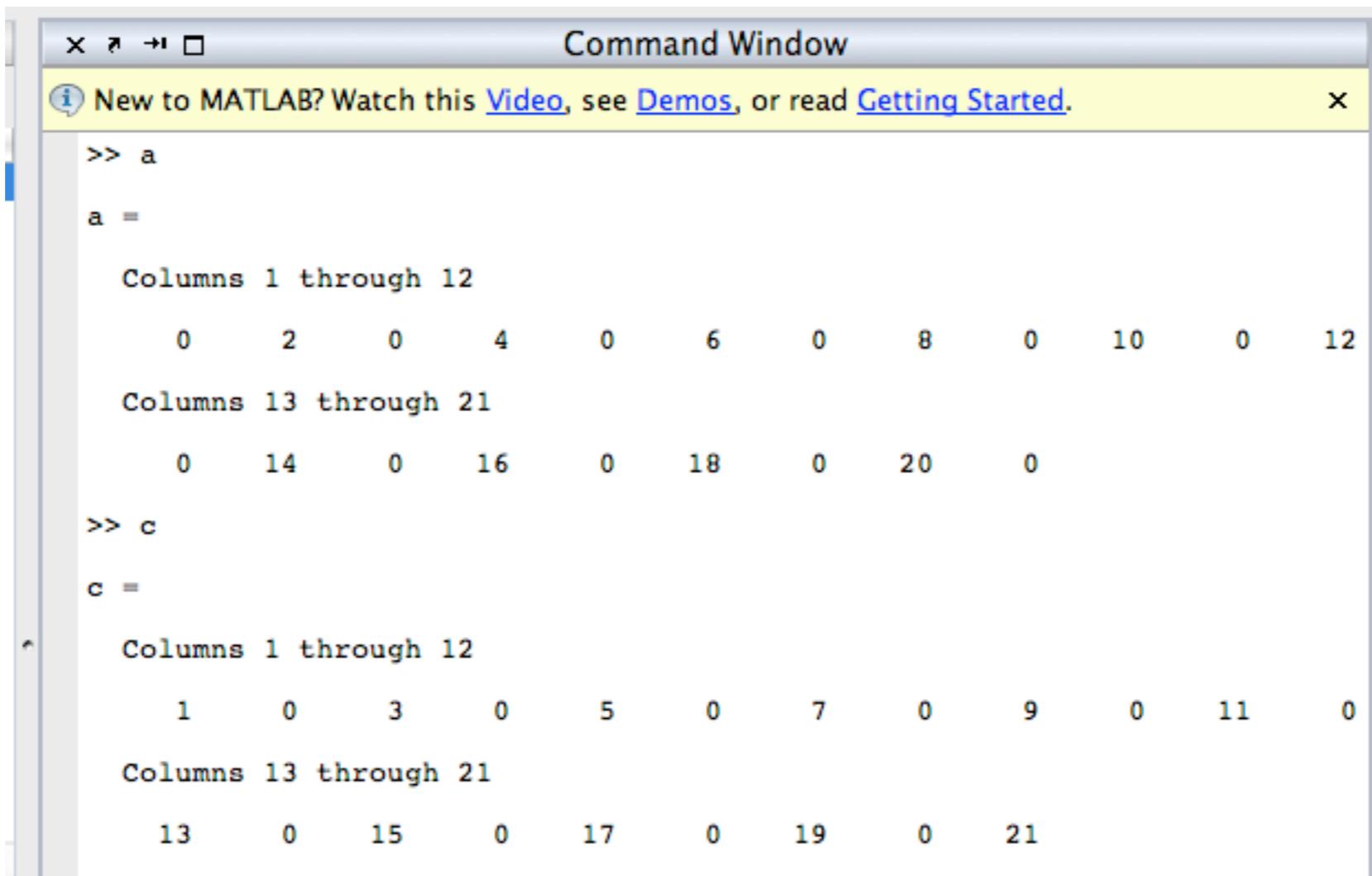
'if' STATEMENT



The screenshot shows the MATLAB Editor interface. The title bar reads "Editor - /Users/macsthil/Documents/MATLAB/ifx.m". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. Below the menu is a toolbar with various icons for file operations like open, save, and print. A numeric toolbar below that shows values 1.0 and 1.1 with operators +, -, *, /, and %.

```
1 - x=1:1:21;
2 - n=max(size(x));
3 - a=zeros(size(x));
4 - c=zeros(size(x));
5 - for k=1:n;
6 -     if floor(x(k)/2)==x(k)/2
7 -         a(k)=x(k);
8 -     else
9 -         c(k)=x(k);
10 -    end
11 - end|
```

OUTPUT OF `if' STATEMENT



The image shows a screenshot of the MATLAB Command Window. The window title is "Command Window". A yellow status bar at the top right says: "New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)". The command input area shows two commands: ">> a" and ">> c". The output for "a" is a 2x12 matrix with values [0, 2, 0, 4, 0, 6, 0, 8, 0, 10, 0, 12] in the first row and [0, 14, 0, 16, 0, 18, 0, 20, 0] in the second row. The output for "c" is a 2x12 matrix with values [1, 0, 3, 0, 5, 0, 7, 0, 9, 0, 11, 0] in the first row and [13, 0, 15, 0, 17, 0, 19, 0, 21] in the second row.

```
>> a
a =
    Columns 1 through 12
    0     2     0     4     0     6     0     8     0    10     0    12
    Columns 13 through 21
    0    14     0    16     0    18     0    20     0
>> c
c =
    Columns 1 through 12
    1     0     3     0     5     0     7     0     9     0    11     0
    Columns 13 through 21
    13    0    15     0    17     0    19     0    21
```

MATLAB PROGRAMMING

M-FILES, SCRIPTS

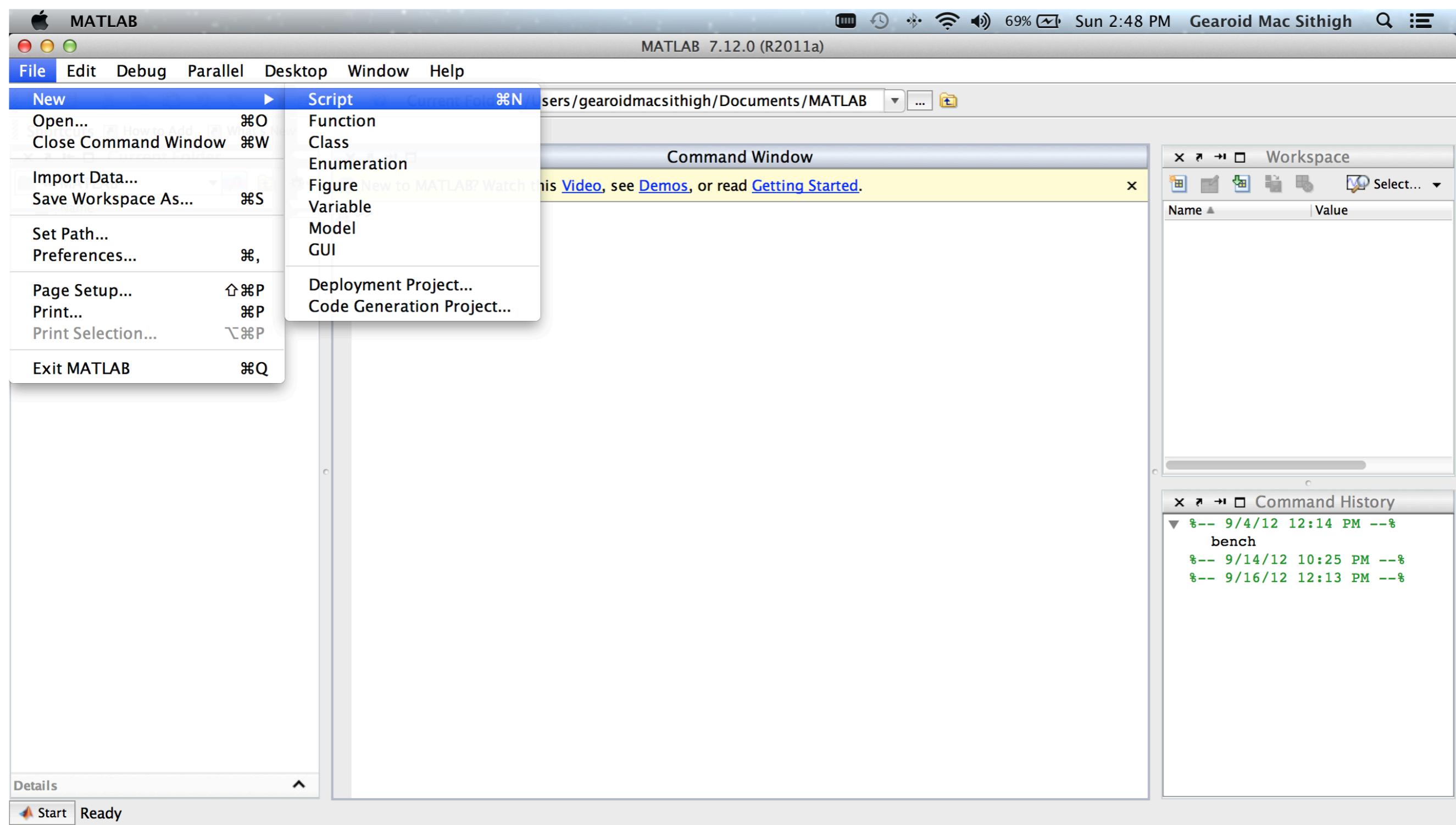
We can't edit or debug in the Command Window.

So, we write any non-trivial piece of code in an M-file. We open the M-file by going to the File pull-down menu; File - New-M-file.

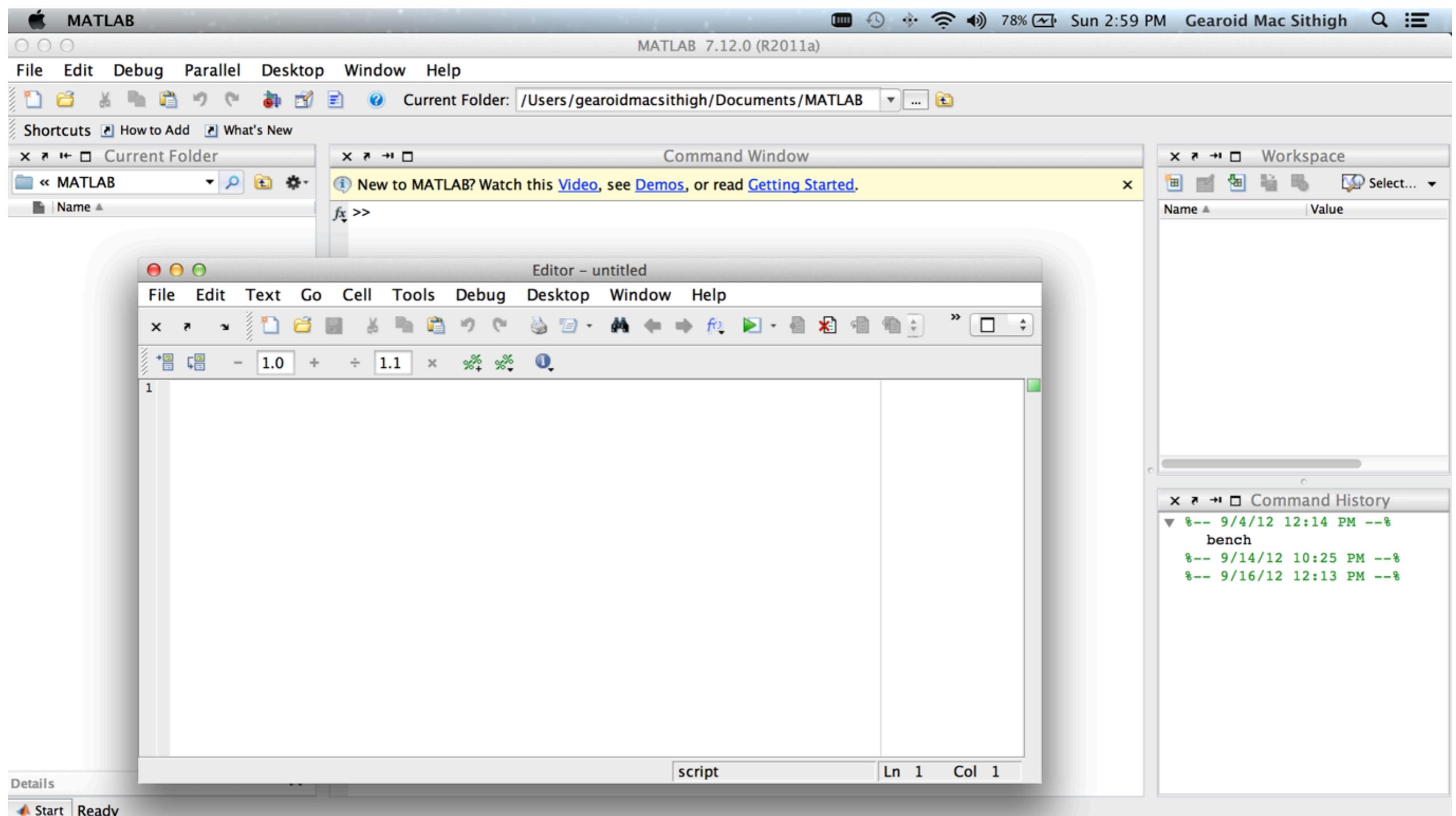
**IN THE LATEST VERSION OF
MATLAB, File-New Script.**

SAVE AS mysterycode.m

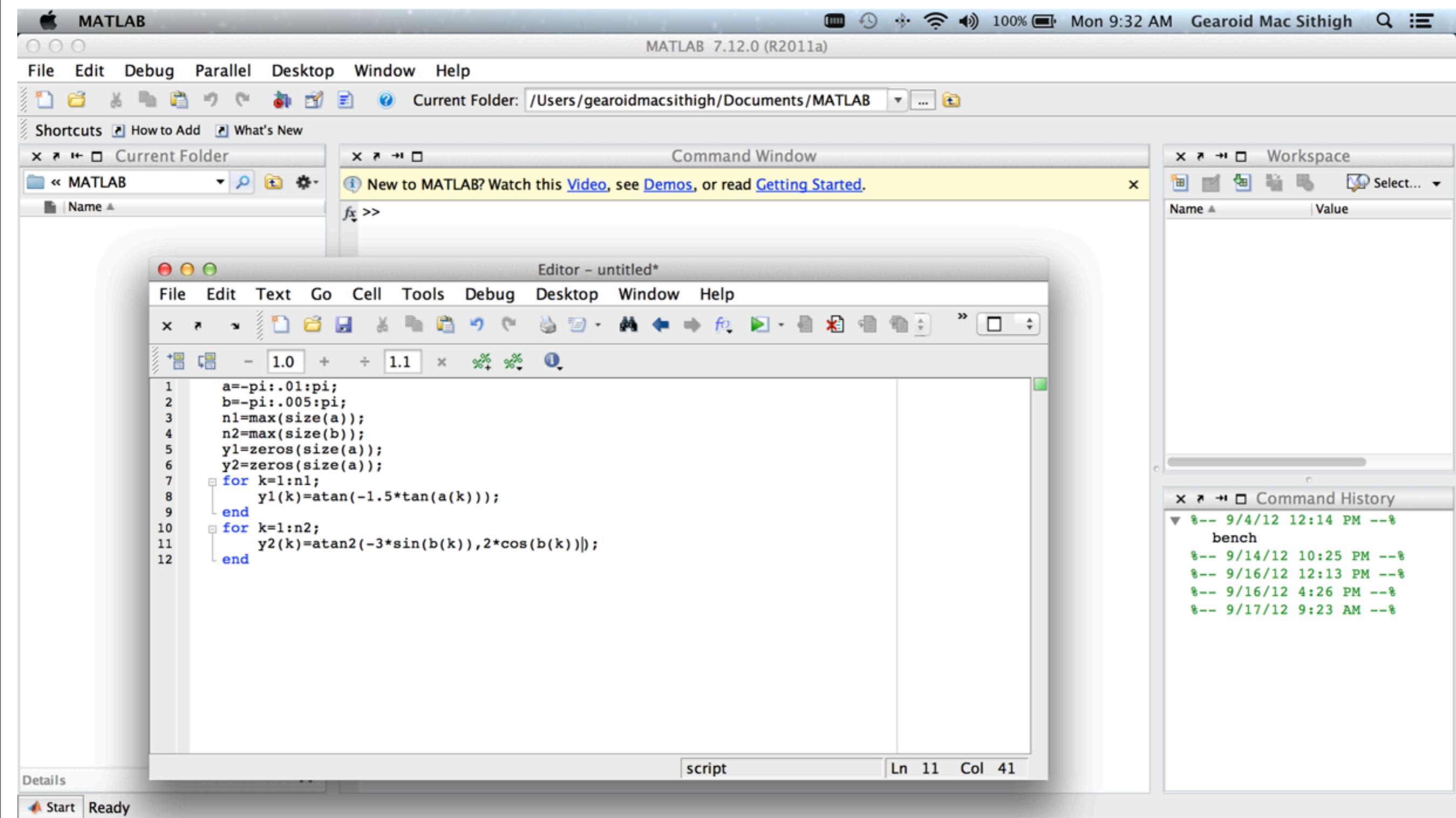
OPENING A SCRIPT



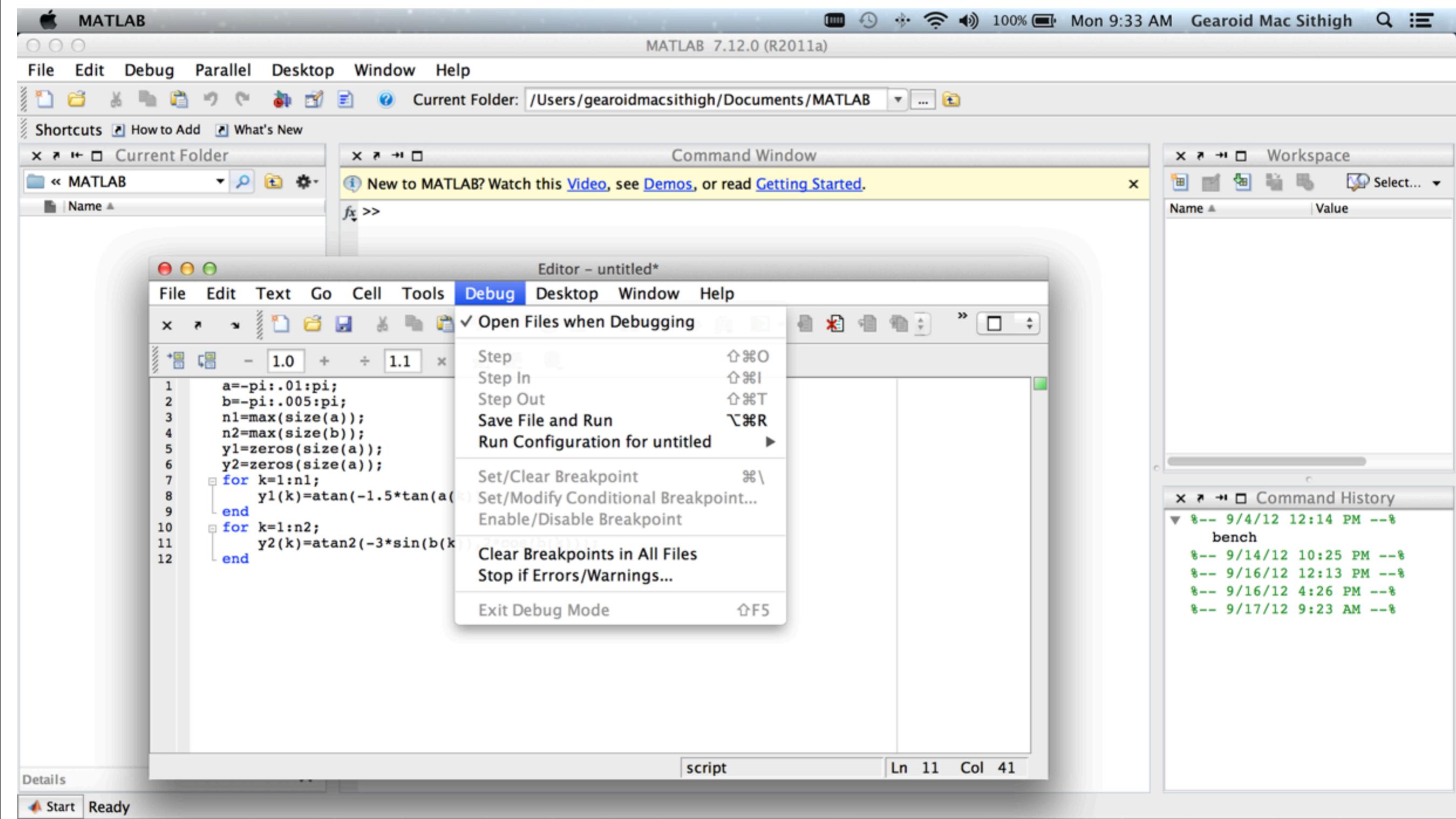
EDITOR WINDOW



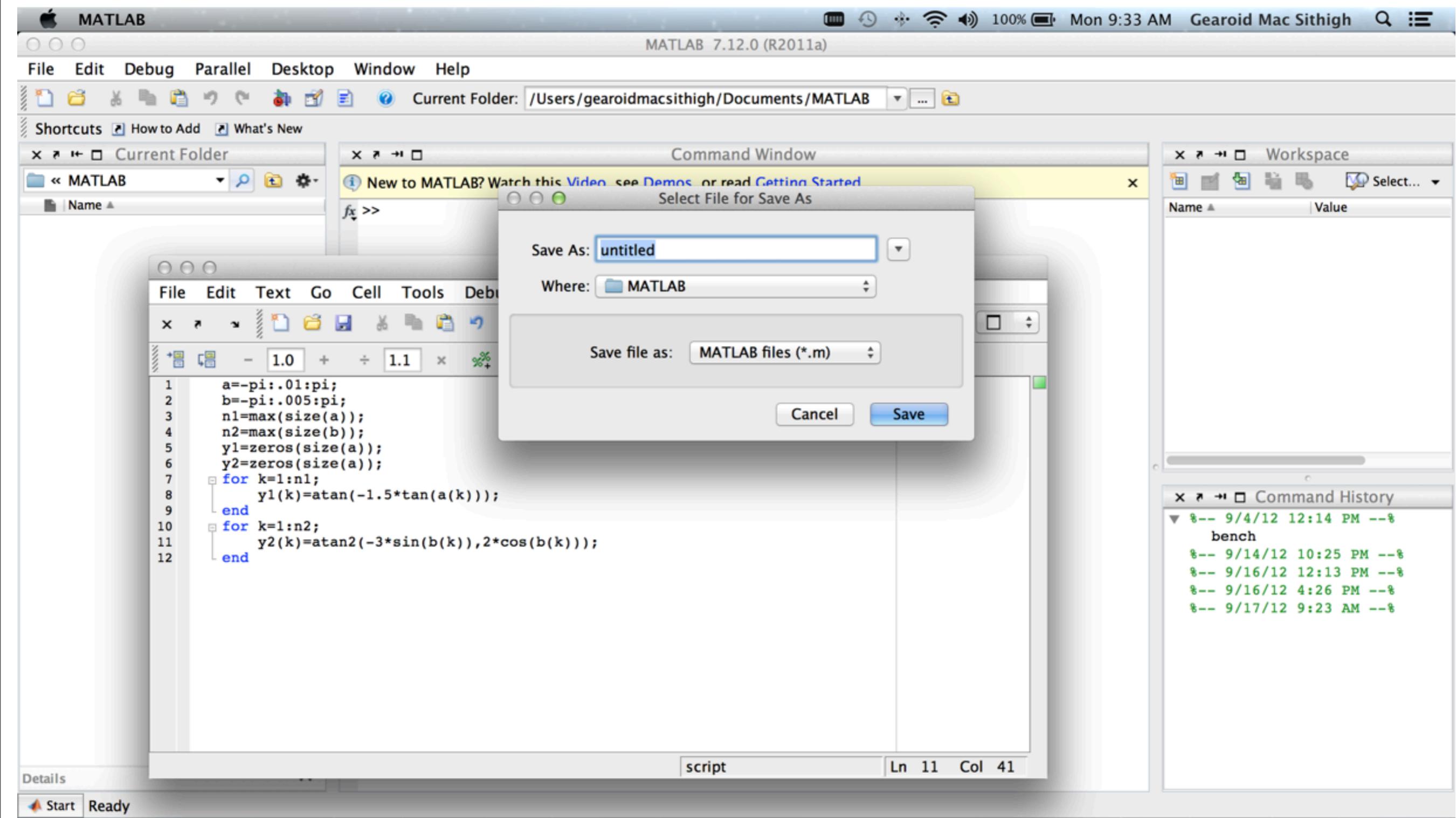
WRITING A SCRIPT



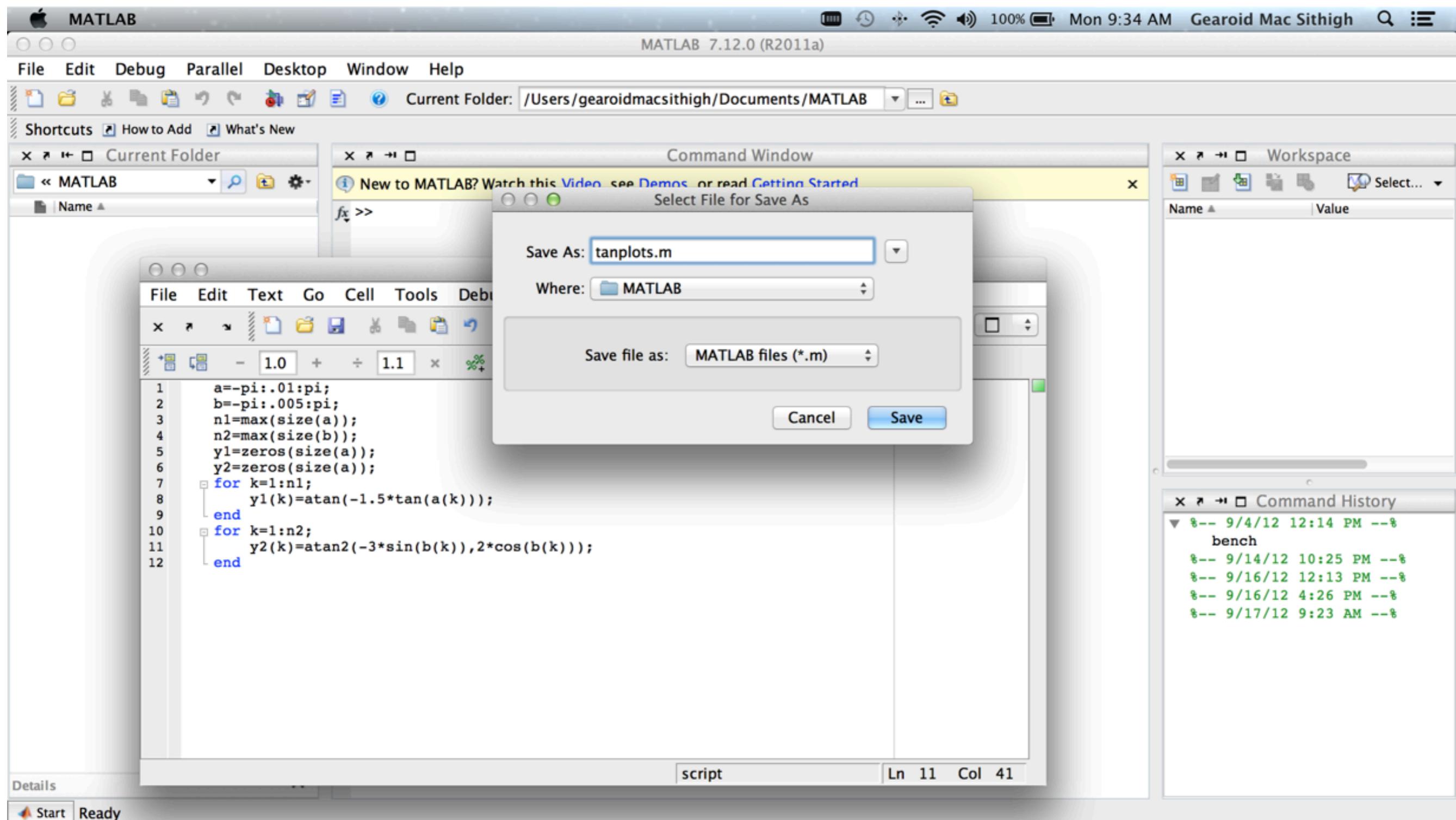
SAVING AND RUNNING A SCRIPT



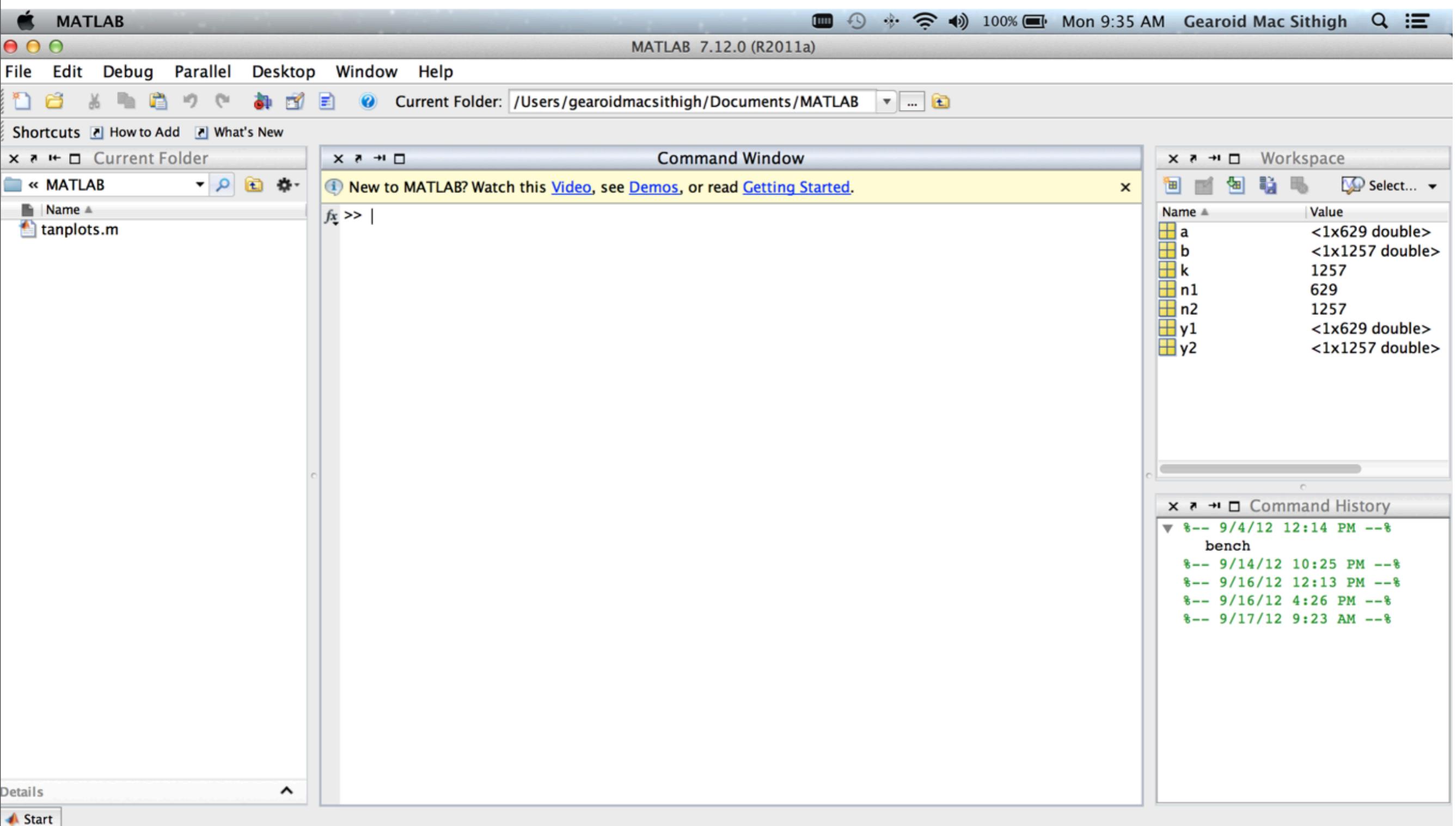
SAVING AND RUNNING A SCRIPT



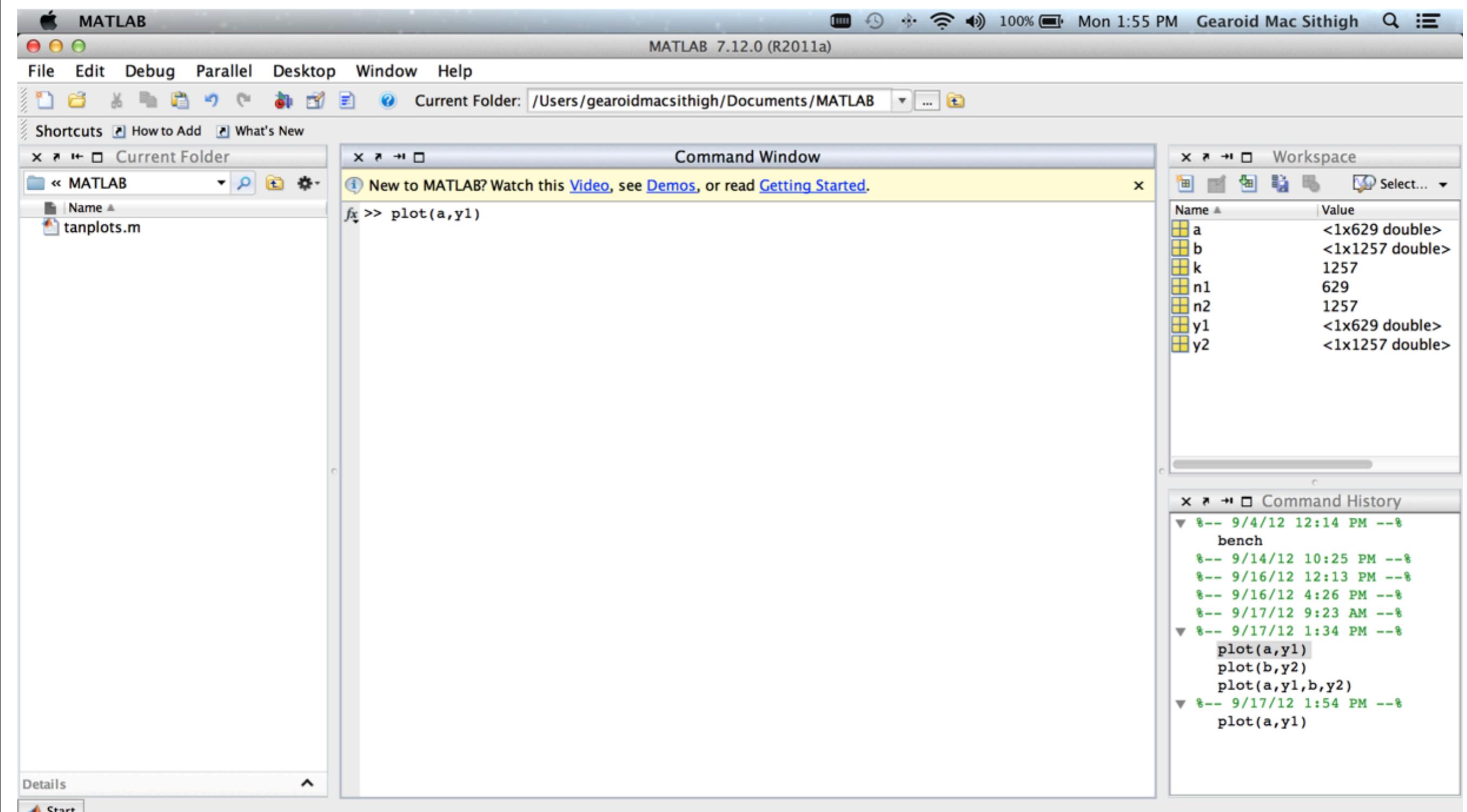
SAVING AND RUNNING A SCRIPT



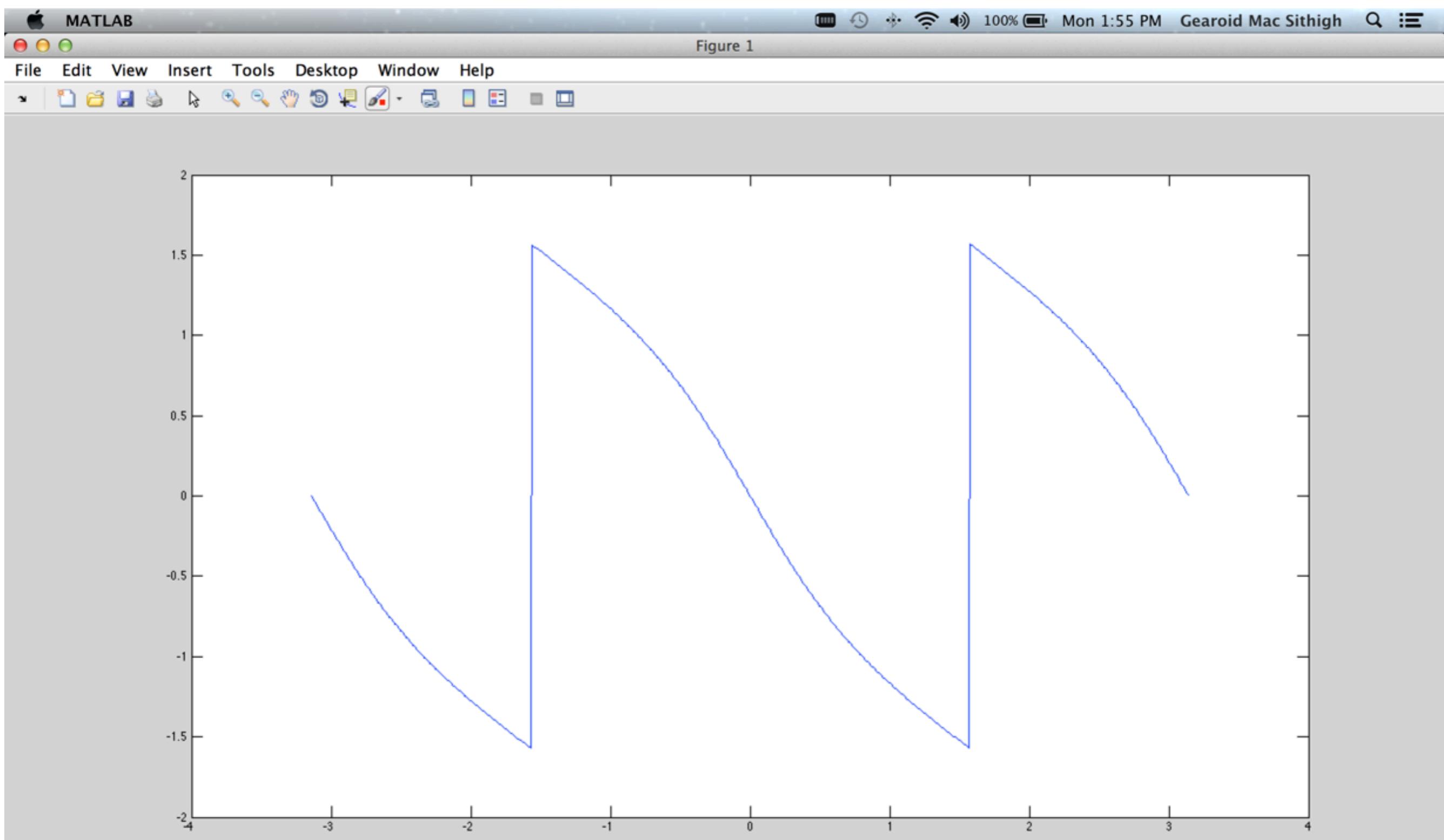
OUTPUT



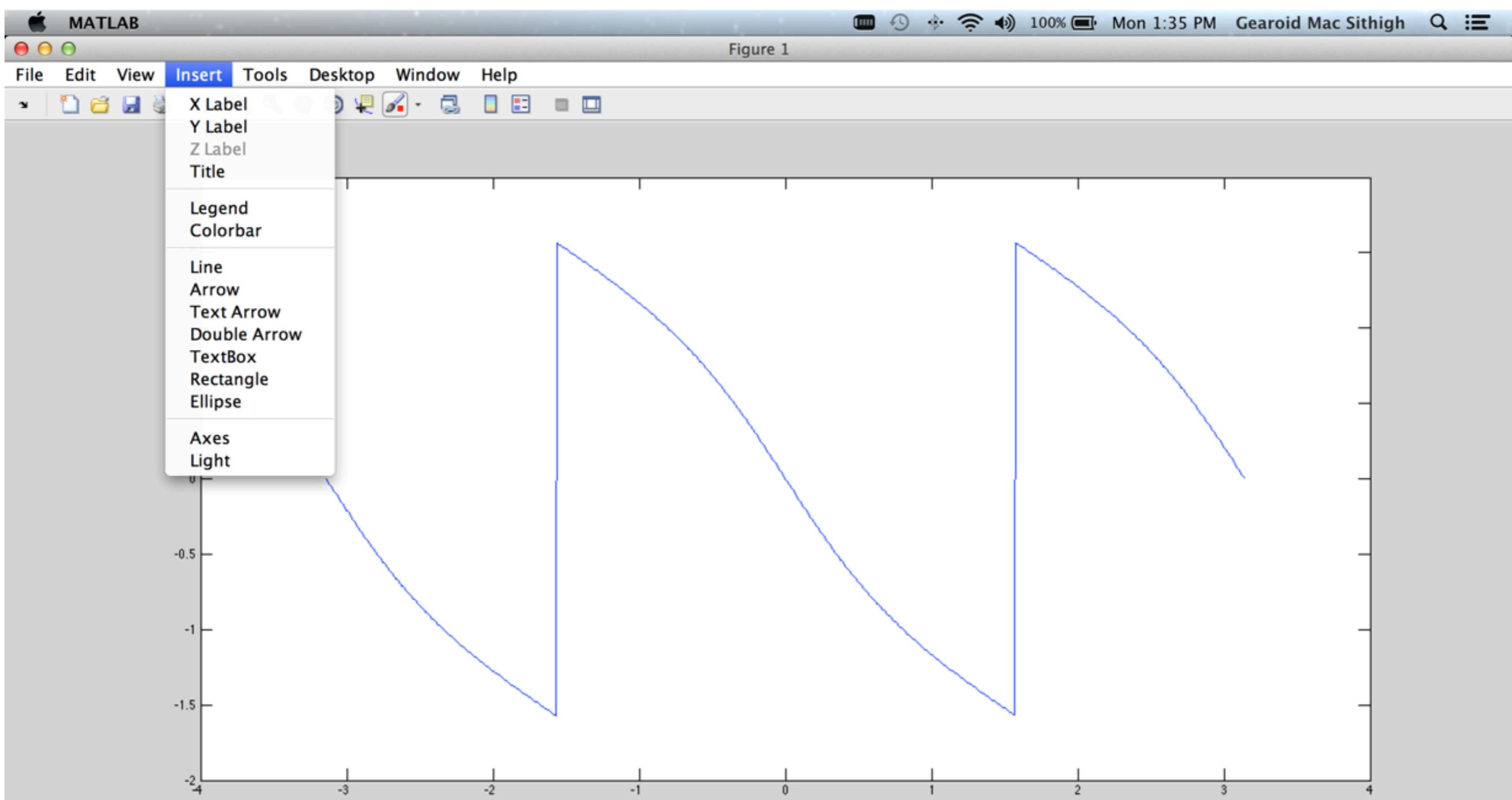
PLOTTING



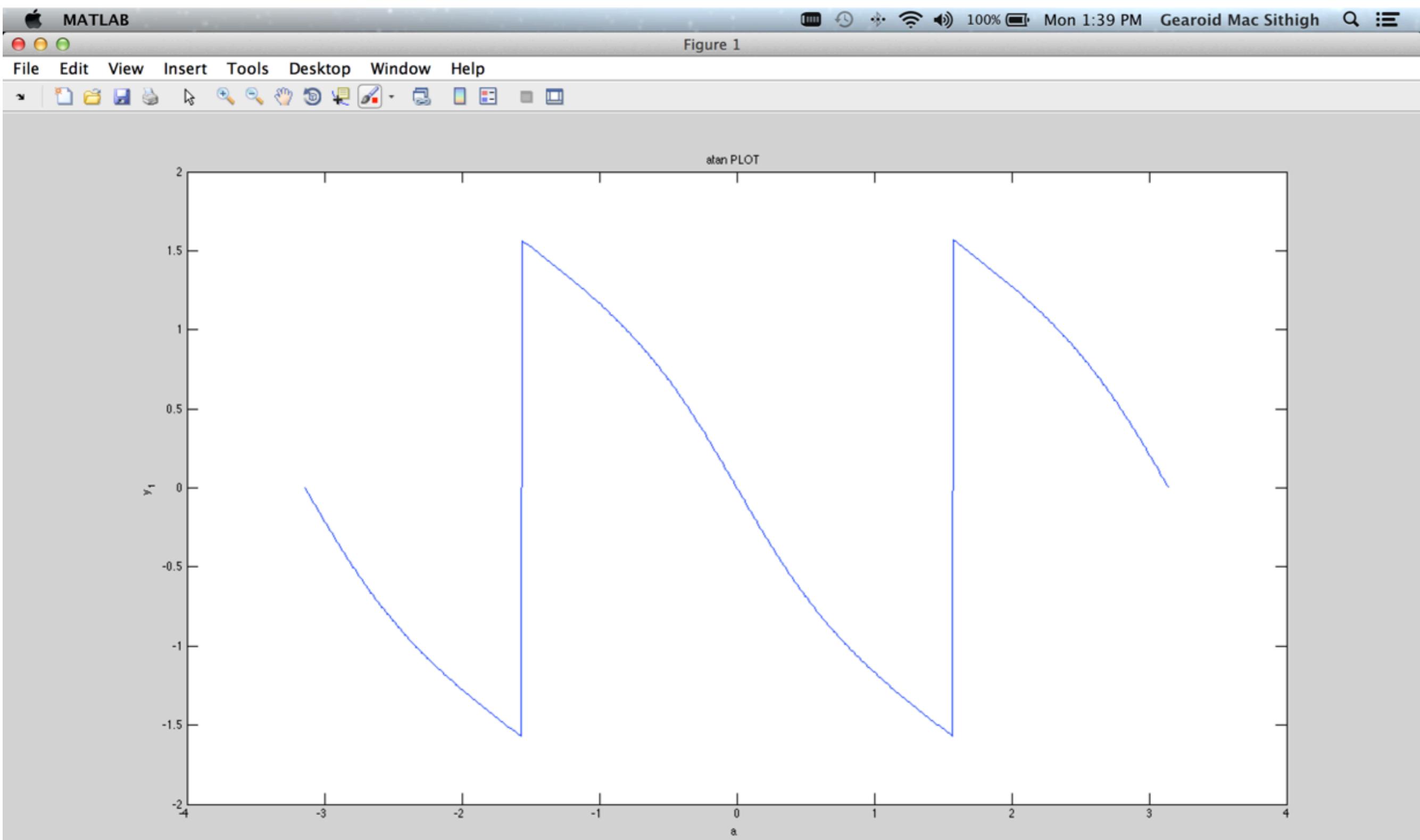
PLOTTING



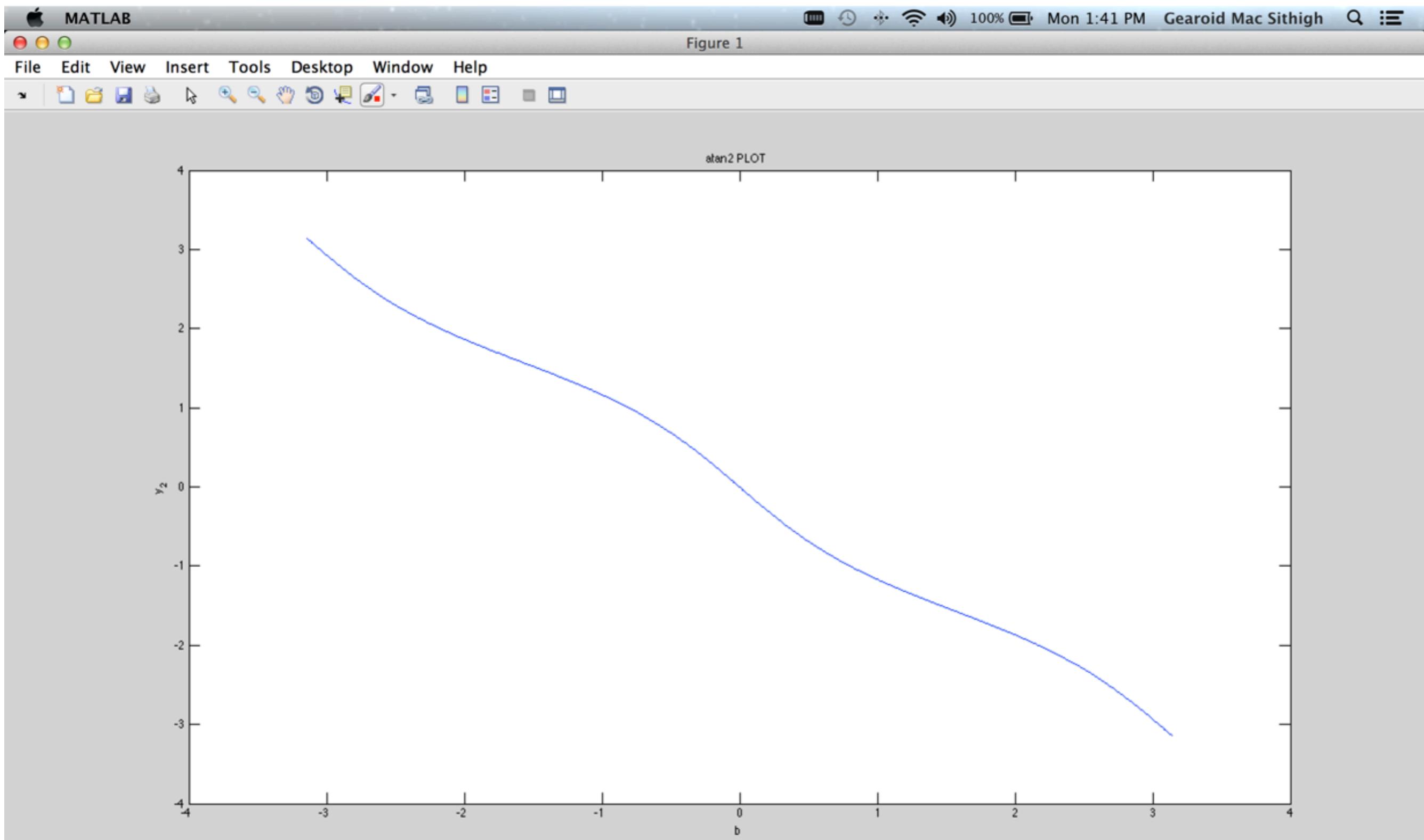
PLOTTING



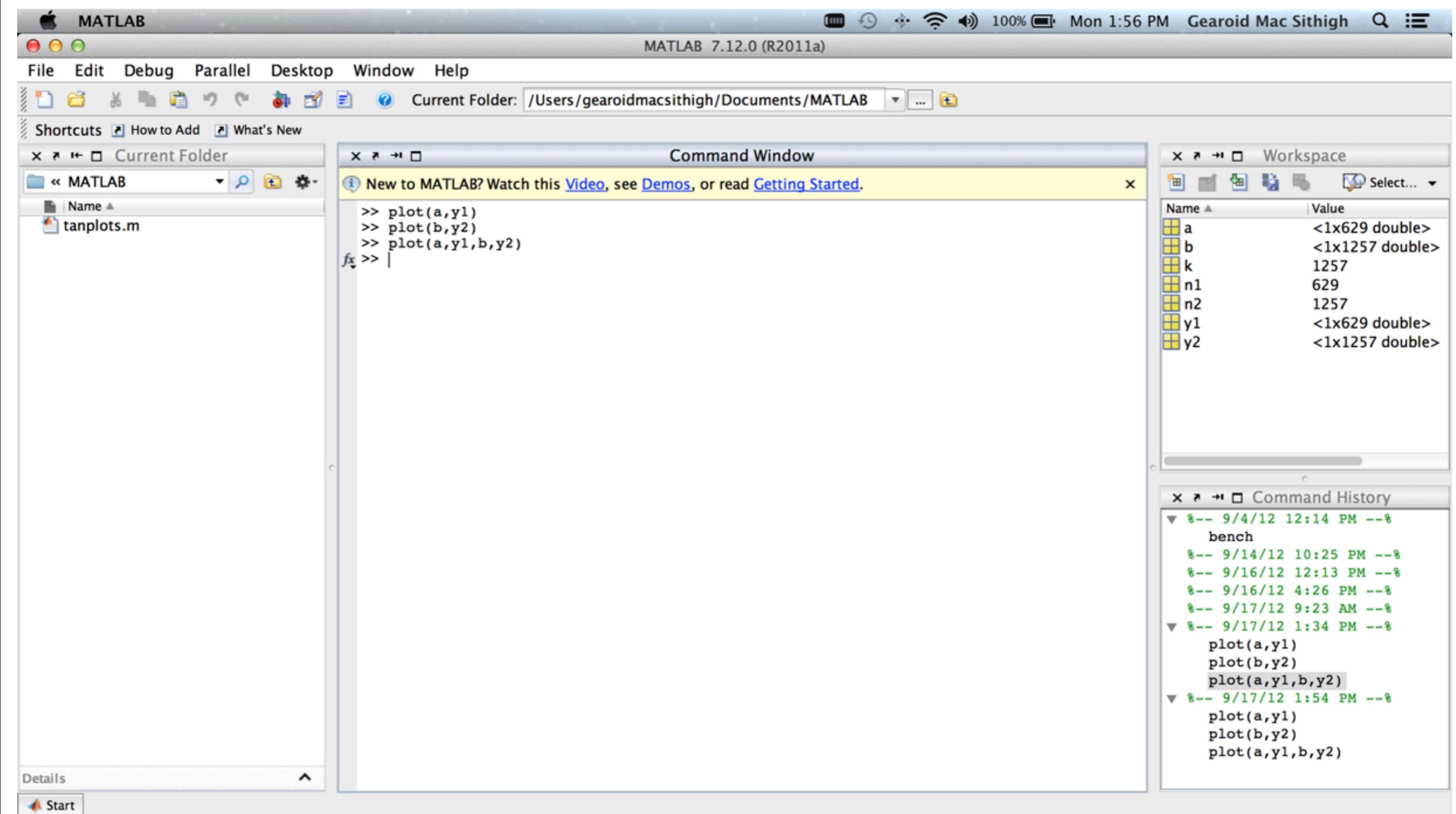
PLOTTING



PLOTTING



PLOTTING



PLOTTING

