

# Chapter 1: An Overview of MATLAB

**MATLAB is:**

A high-level language and interactive environment for numerical computation, visualization, and programming

**MATLAB can:**

Be used as a calculator, easily create scalars, vectors and arrays, be used as a programming environment, make sophisticated plots, be used to create models that describe experimental data, solve statistics and probability problems, solve systems of linear algebraic equations, solve differential equations

**MATLAB is used for:**

Finite element analysis, computational fluid dynamics, signal processing and communications, image and video processing, control systems, test and measurement, etc.

**MATLAB is used by:**

More than a million engineers and scientists in industry and academia

# Chapter 1: An Overview of MATLAB

## Chapter 1 Topics Covered:

- MATLAB Windows: Current Folder, Command, Workspace, Command History
- Using MATLAB as a Calculator
  - Mathematical Operators
  - Clearing Windows
  - Assignment Operator
- Using MATLAB Script Files
  - Creating/Saving/Editing/Executing Script Files
  - Changing the Destination Folder
  - Using the Editor Window
  - Built-In Functions
  - Search Documentation
  - Order of Mathematical Precedence
  - Creating Arrays and using them in Calculations
  - Creating Plots with Arrays (Graphics Window)
  - Creating Arrays Automatically
  - Understanding Error Signals and Error Messages
  - Publishing MATLAB Files

# Using MATLAB as a Calculator

Open MATLAB.

The screenshot shows the MATLAB R2012b desktop environment. The interface includes a top toolbar with icons for various functions, a ribbon menu with tabs for HOME, PLOTS, and APPS, and a search bar for documentation. The main workspace is divided into several panes:

- Current Folder Window:** Located on the left, it shows the current directory path as `C:\Users\scott\Documents\MATLAB`.
- Command Window:** The central pane, containing a prompt `>> |` and a message: "New to MATLAB? Watch this [Video](#), see [Examples](#), or read [Getting Started](#)."
- Workspace Window:** Located on the right, it displays a table with columns for Name and Value, currently empty.
- Command History Window:** Located at the bottom right, it shows a log of previous commands, including the date and time: "11/17/2014 8:17 PM".

Red text labels are overlaid on the image to identify these windows:

- Current Folder Window** (left side)
- Command Window** (center)
- Workspace Window** (right side)
- Command History Window** (bottom right)

The status bar at the bottom left shows "Ready" and the system tray at the bottom right shows "OVR".

# Mathematical Operators

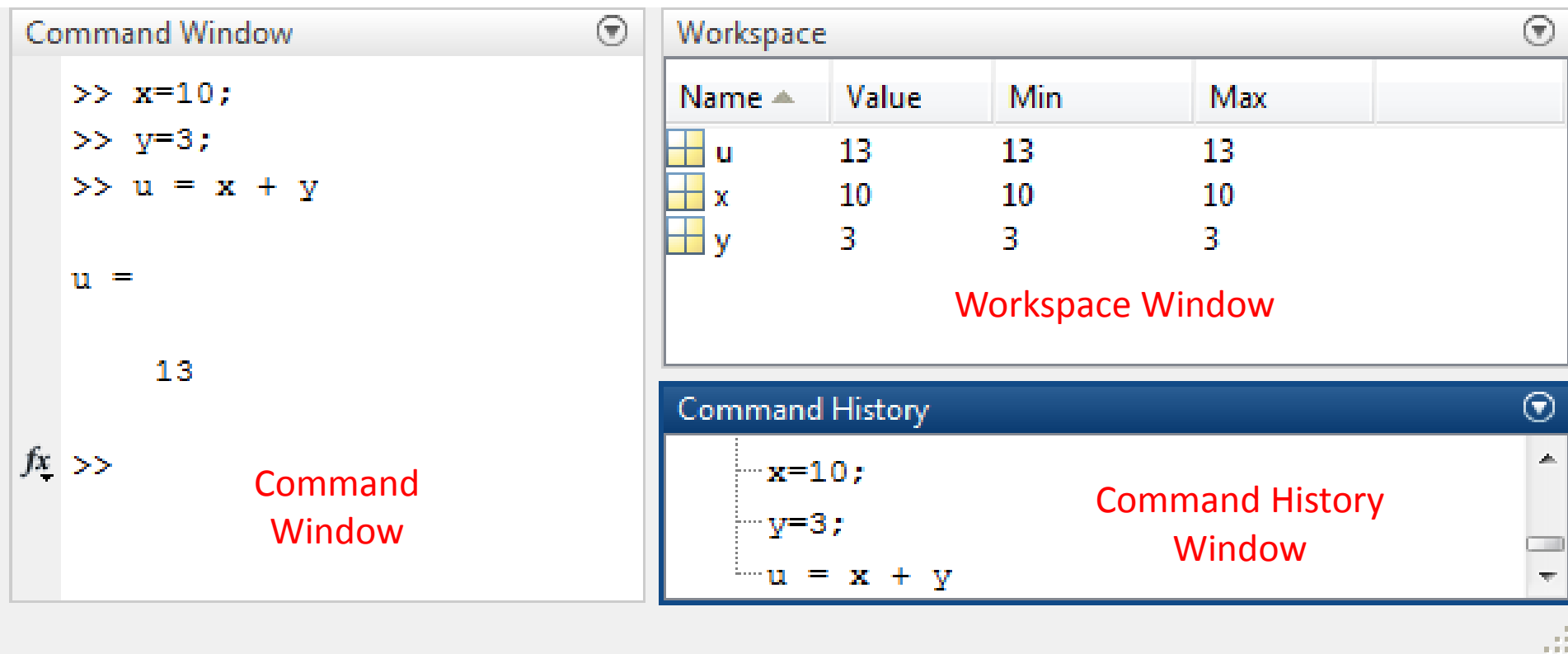
Symbol	Operation	MATLAB form
$\wedge$	exponentiation: $a^b$	$a^b$
$*$	multiplication: $ab$	$a*b$
$/$	right division: $a/b = \frac{a}{b}$	$a/b$
$\backslash$	left division: $a \backslash b = \frac{b}{a}$	$a \backslash b$
$+$	addition: $a + b$	$a+b$
$-$	subtraction: $a - b$	$a-b$

Type the following commands into the **Command Window** to solve Problem 1.1(a):

```
>> x=10;  
>> y=3;  
>> u=x+y
```

1. Make sure you know how to start and quit a MATLAB session. Use MATLAB to make the following calculations, using the values  $x = 10$ ,  $y = 3$ . Check the results by using a calculator.
  - a.  $u = x + y$
  - b.  $v = xy$
  - c.  $w = x / y$
  - d.  $z = \sin x$
  - e.  $r = 8 \sin y$
  - f.  $s = 5 \sin (2y)$

The output of the calculation appears in the **Command Window**.  
The semi-colon (;) suppresses output to the **Command Window**.  
The **Workspace Window** shows the names and values of the **Variables**.  
The **Command History Window** shows the entered commands.  
Previous commands can be accessed quickly by using the **Up Arrow** and **Down Arrow**. Try this!



The screenshot displays three MATLAB windows:

- Command Window:** Shows the execution of commands: `>> x=10;`, `>> y=3;`, and `>> u = x + y`. The output `u =` followed by `13` is visible. A red label "Command Window" is overlaid on the bottom right of this window.
- Workspace:** A table showing the current workspace variables. A red label "Workspace Window" is overlaid on the bottom right of this window.
- Command History:** A list of previously entered commands: `x=10;`, `y=3;`, and `u = x + y`. A red label "Command History Window" is overlaid on the bottom right of this window.

Name ▲	Value	Min	Max
u	13	13	13
x	10	10	10
y	3	3	3

Clear the **Command Window** by typing `clc` (and hit enter) into the **Command Window**.

```
Command Window
>> x=10;
>> y=3;
>> u = x + y

u =

    13

fx >> clc
```

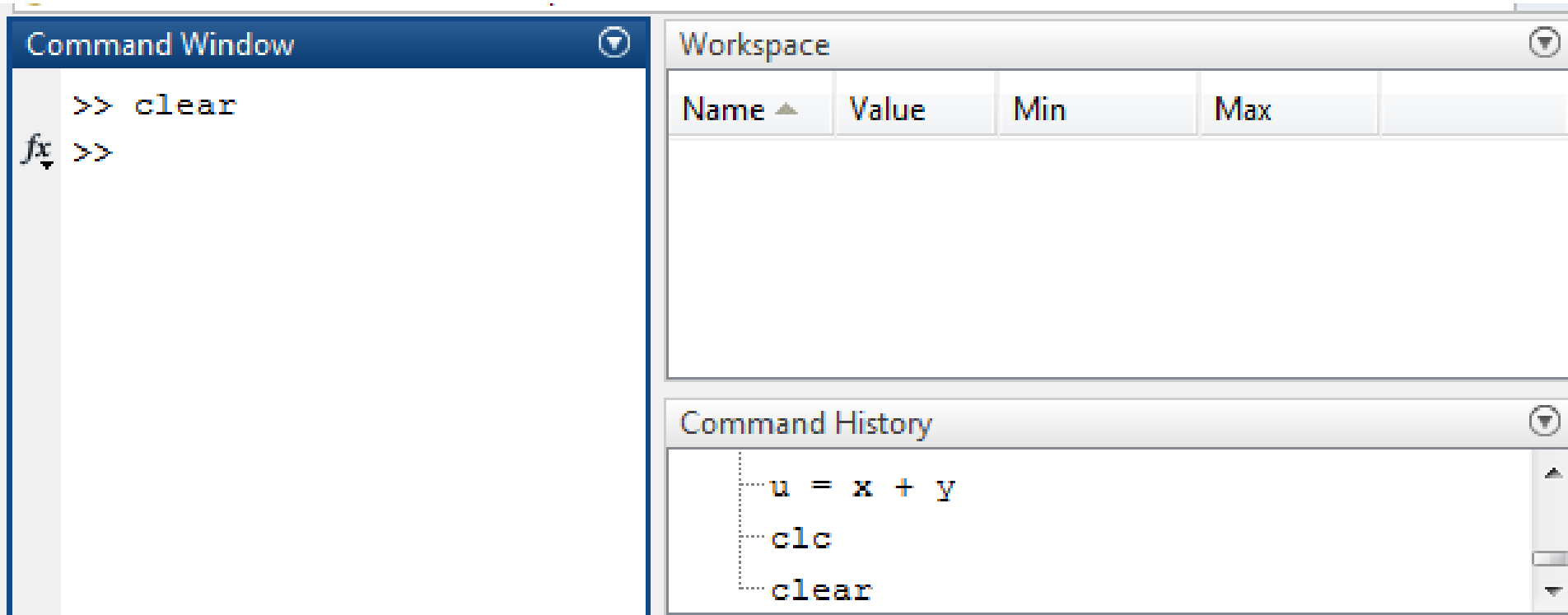
Workspace

Name ▲	Value	Min	Max	
u	13	13	13	
x	10	10	10	
y	3	3	3	

Command History

```
...x=10;
...y=3;
...u = x + y
```

Remove all of the variables from the **Workspace** by typing the command **clear** in the **Command Window**.



The image displays the MATLAB interface with three panels:

- Command Window:** Shows the command `>> clear` being entered. A cursor is visible at the end of the second line, which starts with `fx`.
- Workspace:** A table with columns for Name, Value, Min, and Max. It is currently empty.
- Command History:** A list of commands executed, including `u = x + y`, `clc`, and `clear`.

Name ▲	Value	Min	Max
--------	-------	-----	-----

```
u = x + y
clc
clear
```

In the previous example, the equals sign (=) is called the **Assignment or Replacement Operator**. Type in the following session to demonstrate that the **Assignment Operator** is different than the **equals** sign in mathematics.

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

>> clear
>> x=5

x =

     5

>> x=x+1

x =

     6

>> x=x^2

x =

    36

fx >>
```

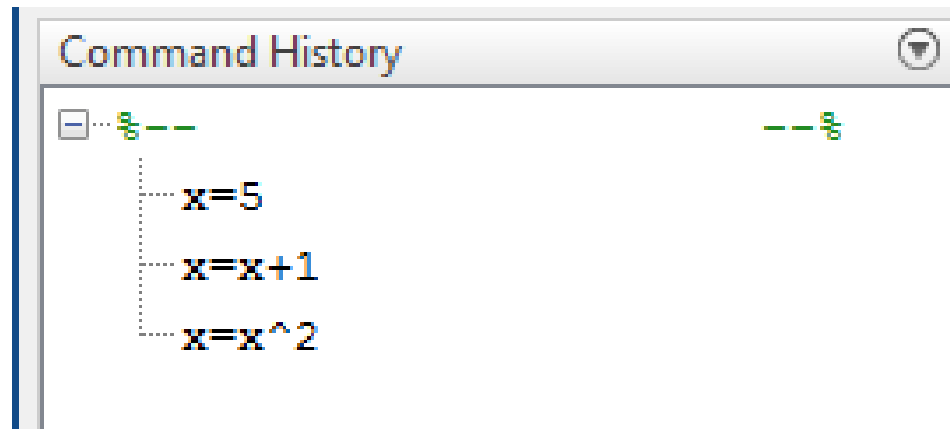


The second command line that was typed in is shown in the **Command History Window** below. It states the following:

“Replace variable **x** with the current contents of variable **x** plus 1.”

In the third command line, the caret symbol (**^**) denotes exponentiation.

“Replace variable **x** with the current contents of variable **x** raised to the second power.”



```
Command History
x=5
x=x+1
x=x^2
```

Use MATLAB to solve Problem 1.3(a) as shown below:

3. Suppose that  $x = 3$  and  $y = 4$ . Use MATLAB to compute the following, and check the results with a calculator.


a.  $\left(1 - \frac{1}{x^5}\right)^{-1}$

b.  $3\pi x^2$

c.  $\frac{3y}{4x - 8}$

d.  $\frac{4(y - 5)}{3x - 6}$

## Command Window

 New to MATLAB? Watch this [Video](#),

```
>> x=3;
```

```
>> u=(1-1/x^5)^(-1)
```

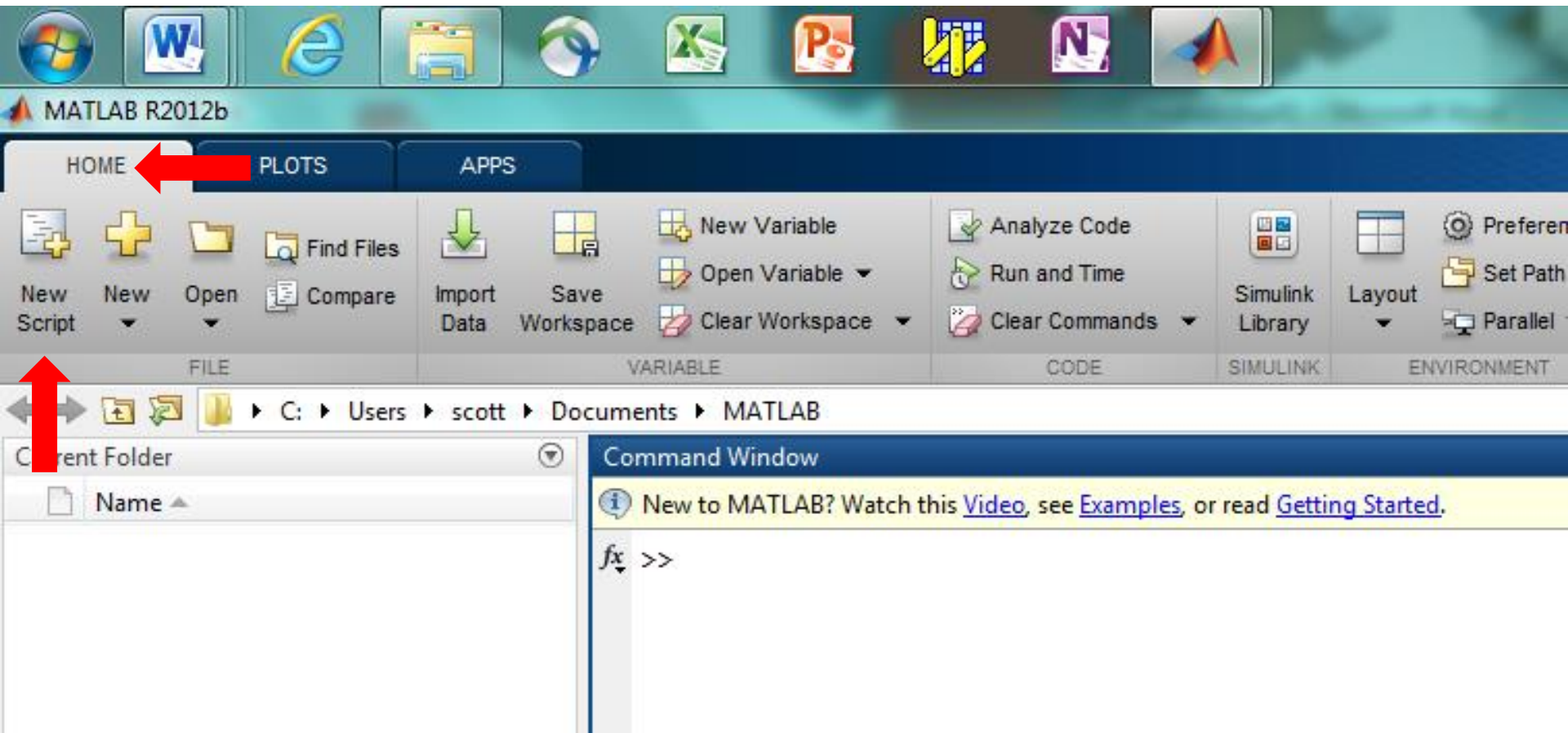
```
u =
```

```
1.0041
```

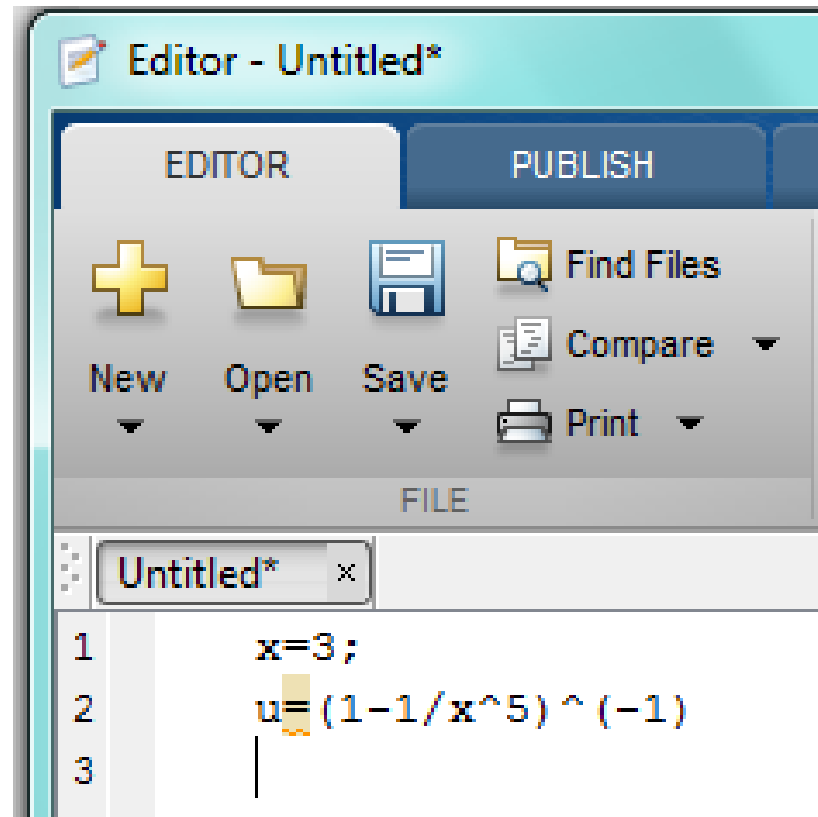
```
fx >> |
```

# Using MATLAB Script Files

Another method of computation is to create a **Script File**, which is a way to store commands to be executed in the **Command Window**. Use the **Home/New Script** tab to create a new **Script File**:



The new window that has appeared is called the **Editor/Debugger**. Use the **Editor** in MATLAB to solve Problem 1.3(a) by typing the following.



3. Suppose that  $x = 3$  and  $y = 4$ . Use MATLAB to compute the following, and check the results with a calculator.

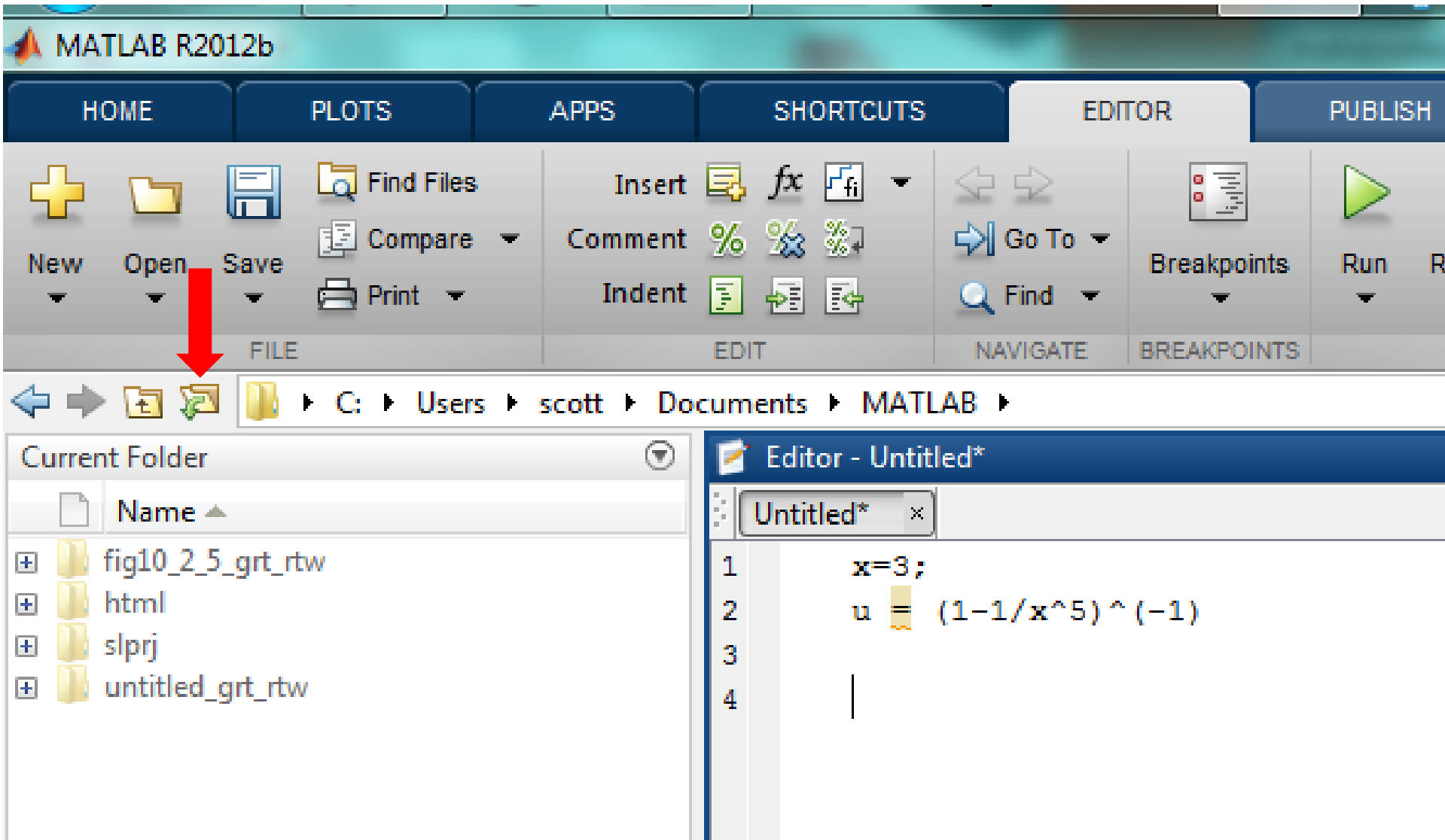
a.  $\left(1 - \frac{1}{x^5}\right)^{-1}$

b.  $3\pi x^2$

c.  $\frac{3y}{4x - 8}$

d.  $\frac{4(y - 5)}{3x - 6}$

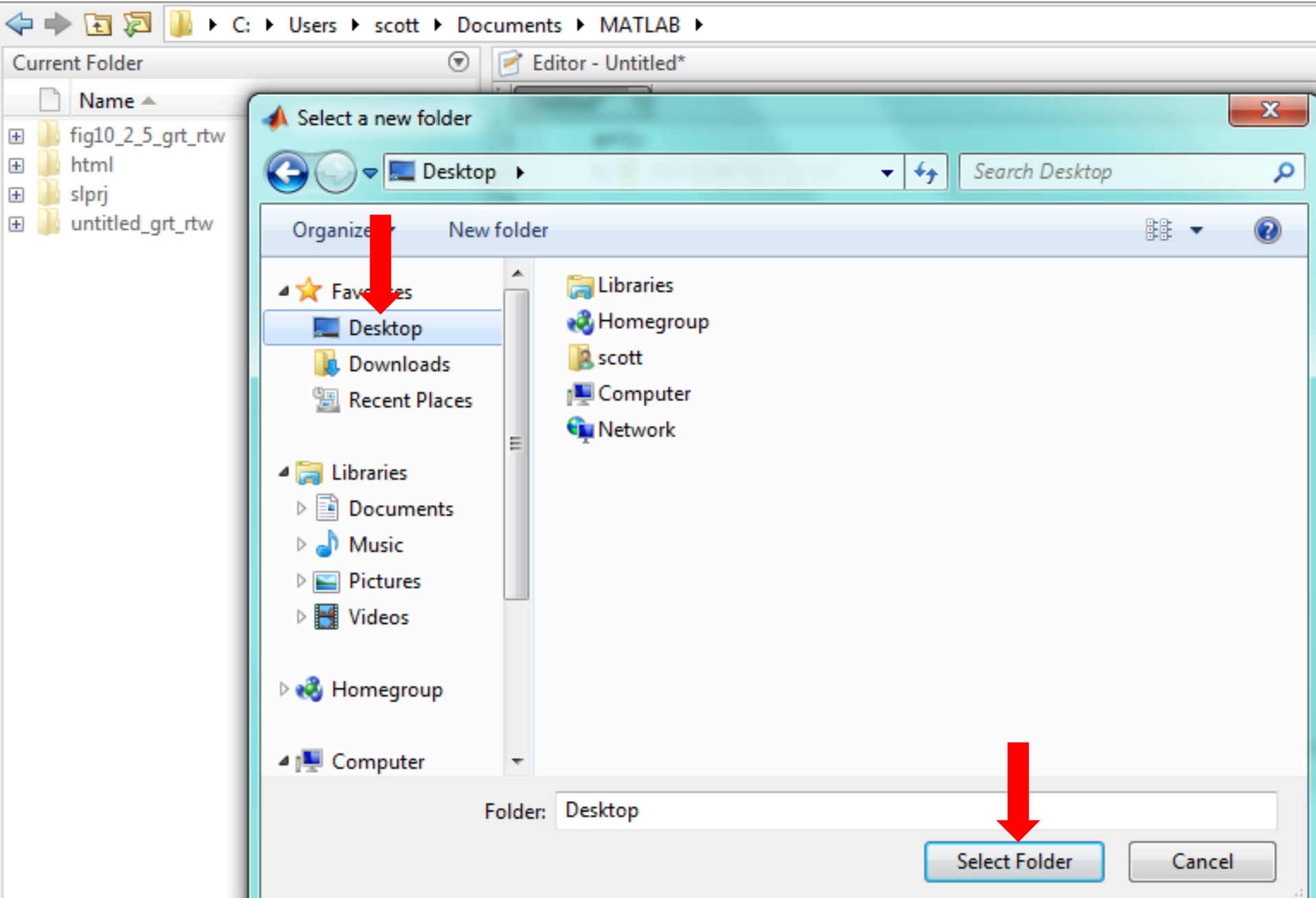
Before saving, change the folder that MATLAB saves files to (the **Destination Folder**) by pushing the **Browse for Folder** button:



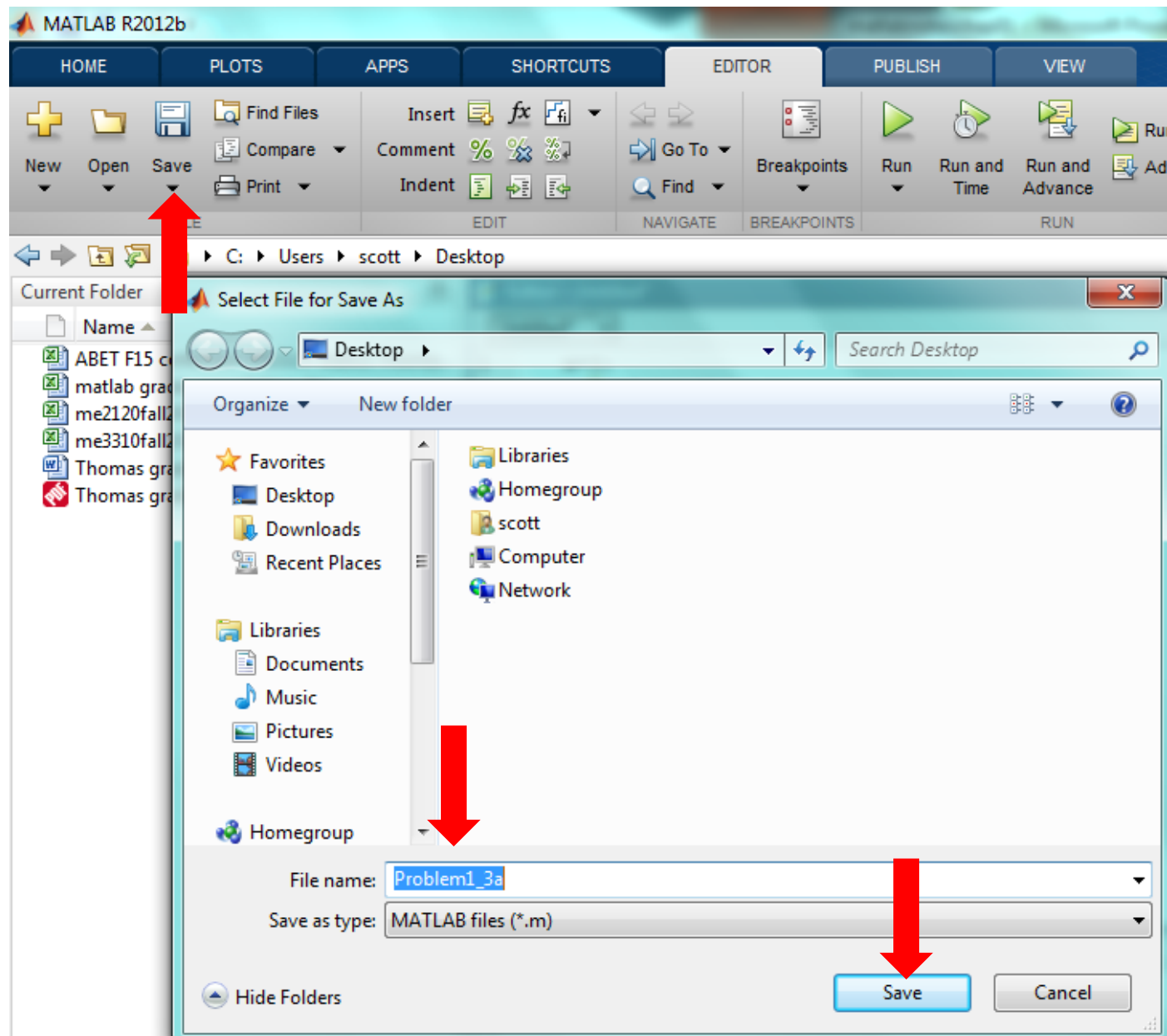
The image shows the MATLAB R2012b interface. The top menu bar includes HOME, PLOTS, APPS, SHORTCUTS, EDITOR, and PUBLISH. The EDITOR tab is active, showing a toolbar with various icons. A red arrow points to the 'Open' button in the FILE section of the toolbar. Below the toolbar, the breadcrumb path is C:\Users\scott\Documents\MATLAB. The 'Current Folder' pane on the left shows a list of folders: fig10\_2\_5\_grt\_rtw, html, slprj, and untitled\_grt\_rtw. The 'Editor - Untitled\*' window on the right shows a code editor with the following code:

```
1     x=3;  
2     u = (1-1/x^5)^(-1)  
3  
4
```

Select the **Desktop** folder to be the **Destination Folder**.



Save the file using the **Save** button. MATLAB **Script File** names must start with a letter, and the only special character allowed is the underscore. Numbers are allowed, as long as the number is not the first character. Spaces are not allowed.



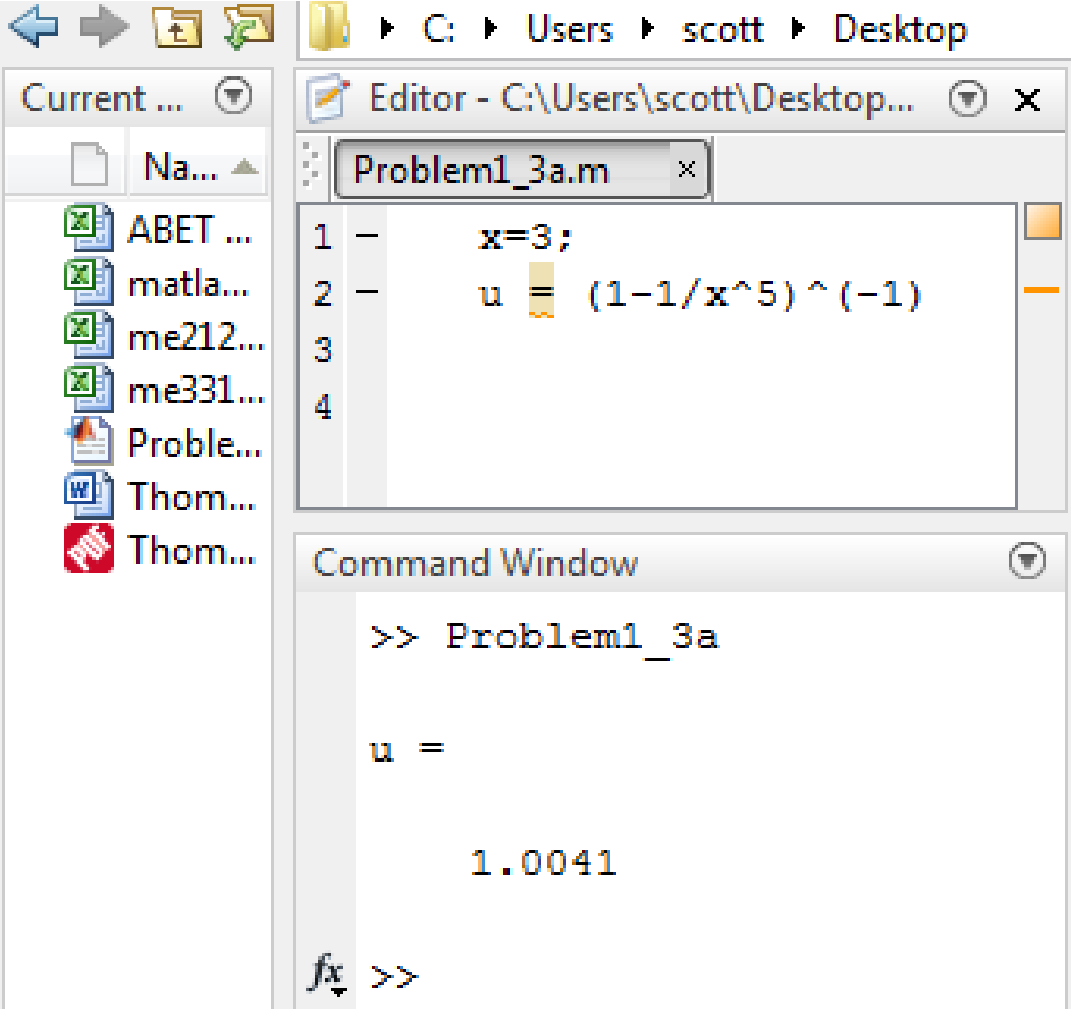
The file appears in the Desktop folder as a **\*.m file**. The **Script File** must be saved prior to execution. Press the green **Run** button to execute the Script file. Alternatively, pressing the **Run** button will automatically save the **Script File**.

The image shows the MATLAB R2012b software interface. The top menu bar includes HOME, PLOTS, APPS, SHORTCUTS, EDITOR, and PUBLISH. The ribbon below contains several groups of icons: FILE (New, Open, Save, Find Files, Compare, Print), EDIT (Insert, Comment, Indent), NAVIGATE (Go To, Find), BREAKPOINTS, and a Run button (a green play icon) which is highlighted with a red arrow. Below the ribbon, the current folder is C:\Users\scott\Desktop. The file explorer shows a list of files, with 'Problem1\_3a.m' highlighted by a red arrow. The editor window shows the script content:

```
1 - x=3;  
2 - u = (1-1/x^5)^(-1)  
3  
4 |
```



The results of the calculation appear in the **Command Window**. Notice that the values shown in the **Workspace Window** have changed.



The image shows the MATLAB Editor window with the following content:

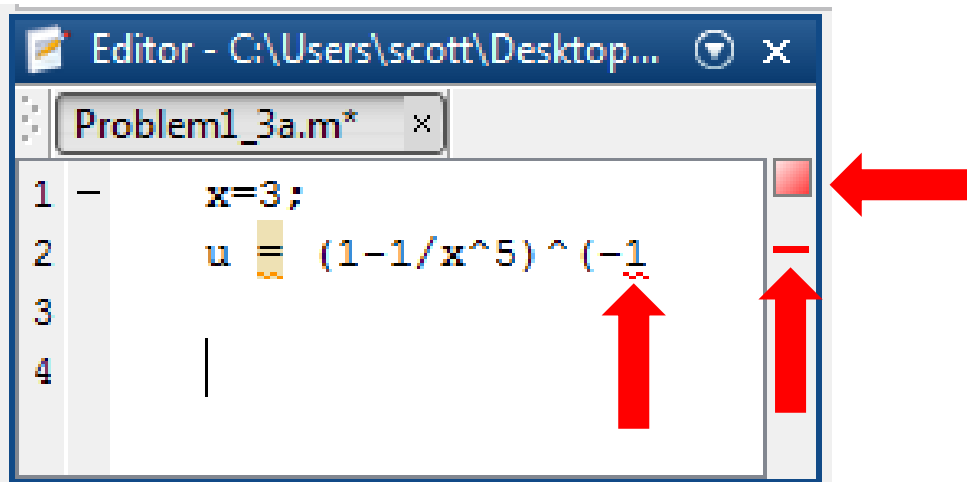
```
1 - x=3;  
2 - u = (1-1/x^5)^(-1)  
3  
4
```

Below the editor is the Command Window with the following text:

```
>> Problem1_3a  
  
u =  
  
    1.0041  
  
fx >>
```

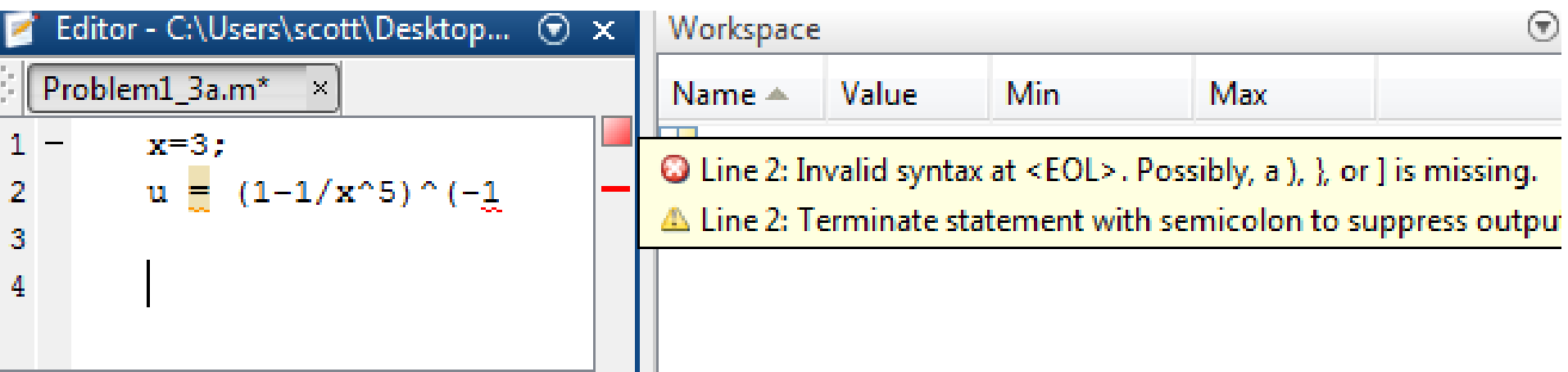
Name	Value	Min	Max
u	1.0041	1.0041	1.0041
x	3	3	3

MATLAB will give you clues when you make a mistake in your equations. Delete the last parenthesis in the equation. Notice that the square becomes red, and a red line appears at the line that has the mistake. Also, a squiggly line appears where MATLAB thinks the error lies.



```
1 - x=3;
2 u = (1-1/x^5)^(-1;
3 |
4 |
```

Hover the cursor over the red line: EOL stands for End Of Line.



```
1 - x=3;
2 u = (1-1/x^5)^(-1;
3 |
4 |
```

Name	Value	Min	Max
------	-------	-----	-----

Line 2: Invalid syntax at <EOL>. Possibly, a ), }, or ] is missing.

Line 2: Terminate statement with semicolon to suppress output.

Make the following change in the **Script File** and save the program using the **Save Button** under the **Editor** drop-down menu prior to running the program.

```
Problem1_3a.m ×
1 - x=3;
2 - u = (1-1/x^5)^(-1)
3 - w = 3*pi*x^2
4
5
6
```

```
Command Window
u =
    1.0041

w =
    84.8230

fx >>
```

Name ▲	Value	Min	Max	
u	1.0041	1.0041	1.0041	
w	84.8230	84.8230	84.8230	
x	3	3	3	

```
Command History
...x=10;
...y=3;
...u = x + y
...clc
...clear
...Problem1_3a
```

In the modified **Script File**, the built-in variable **pi** was used. Other built-in functions exist, as shown in the following session where Problem 1.16(a) is to be solved.

Function	MATLAB syntax <sup>1</sup>
$e^x$	exp(x)
$\sqrt{x}$	sqrt(x)
$\ln x$	log(x)
$\log_{10} x$	log10(x)
$\cos x$	cos(x)
$\sin x$	sin(x)
$\tan x$	tan(x)
$\cos^{-1} x$	acos(x)
$\sin^{-1} x$	asin(x)
$\tan^{-1} x$	atan(x)

<sup>1</sup>The MATLAB trigonometric functions use radian measure.

**16.** Use MATLAB to calculate

a.  $6\pi \tan^{-1}(12.5) + 4$

b.  $5 \tan [3 \sin^{-1}(13/5)]$

c.  $5 \ln(7)$

d.  $5 \log(7)$

Check your answers with a calculator.

Create a new **Script File** using the **New File Button** on the **Editor**. MATLAB gives the new file the temporary name **Untitled2**.

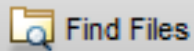
EDITOR

PUBLISH

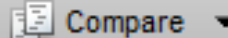
VIEW



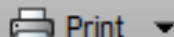
New Open Save



Find Files



Compare



Print

FILE

EDIT

NAVIGATE



Breakpoints

BREAKPOINTS



Run



Run and Time



Run and Advance



Run Section



Advance

RUN

Problem1\_3a.m

```
1 - x=3;  
2 - u=(1-1/x^5)^(-1)  
3 - w=3*pi*x^2  
4 |
```

script

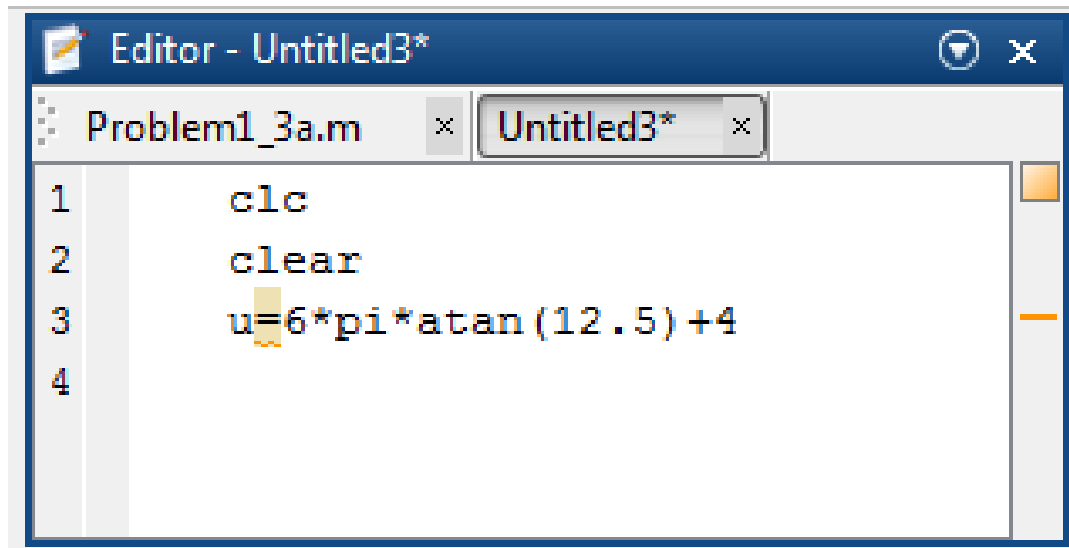
Ln 4

Col 1

OVR

Type in the following to calculate the equation given in Problem 1.16(a). Get in the habit of typing in the first two commands to clear the command window and to clear the variables stored in memory.

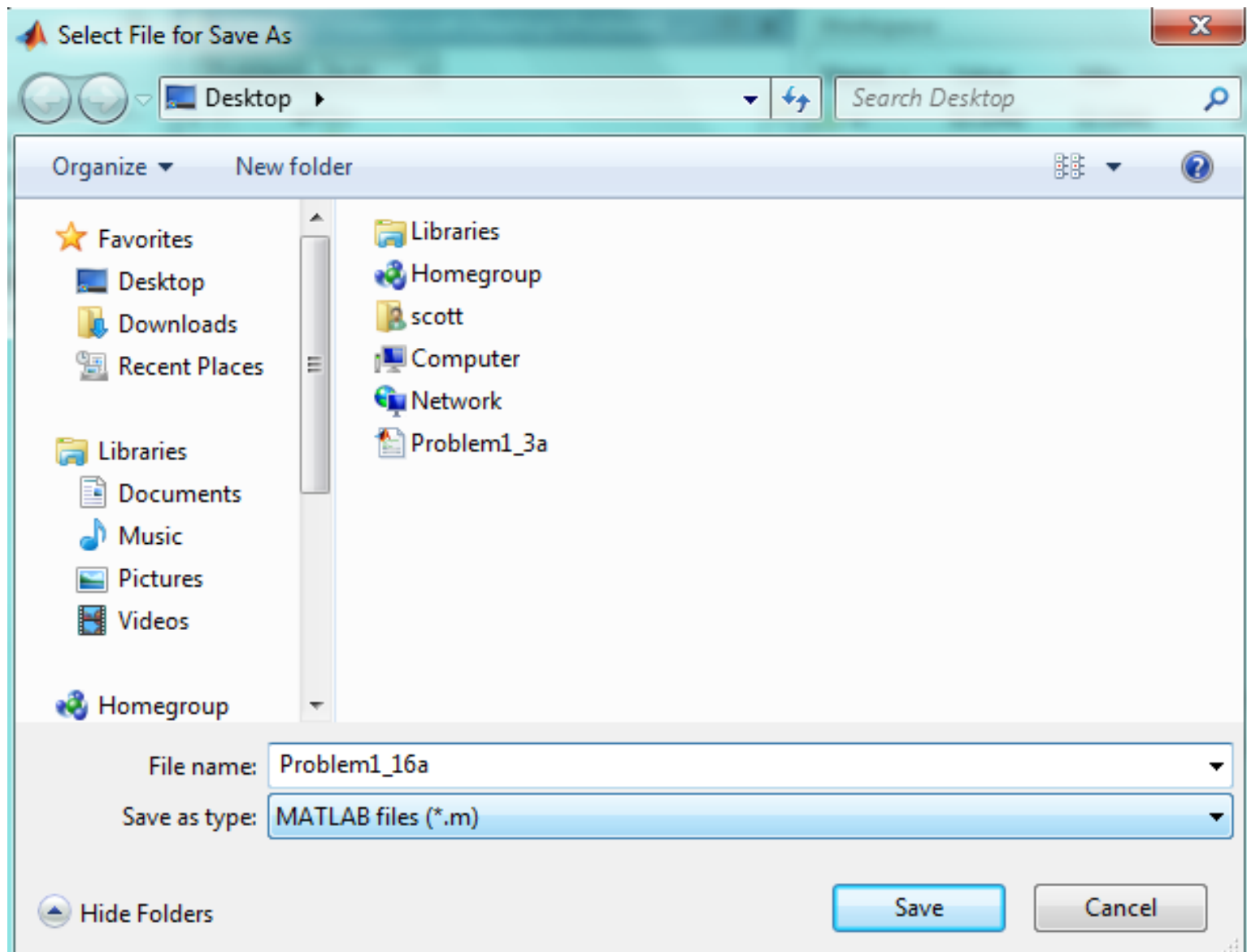
$$6\pi \tan^{-1}(12.5) + 4$$



The screenshot shows a MATLAB Editor window titled "Editor - Untitled3\*". The window contains two tabs: "Problem1\_3a.m" and "Untitled3\*". The code in the "Untitled3\*" tab is as follows:

```
1      clc
2      clear
3      u=6*pi*atan(12.5)+4
4
```

Save it to the Desktop using the **Save As** drop-down menu:



Notice that you can toggle back and forth between the two saved files by clicking on the names in the **Editor Window**. Try it! Press the **Run** button to execute the new **Script File**.

Current Folder

Name
ABET F15 collection ME...
matlab grade compilati...
me2120fall2015.xlsx
me3310fall2015.xlsx
<b>Problem1_3a.m</b>
Problem1_16a.m
Thomas grades fall 2015...
Thomas grades fall 2015...

Editor - C:\Users\scott\Desktop\problem1\_16a.m

Problem1\_3a.m × Problem1\_16a.m ×

```
1 -   clc
2 -   clear
3 -   u = 6*pi*atan(12.5) + 4
4
5
```

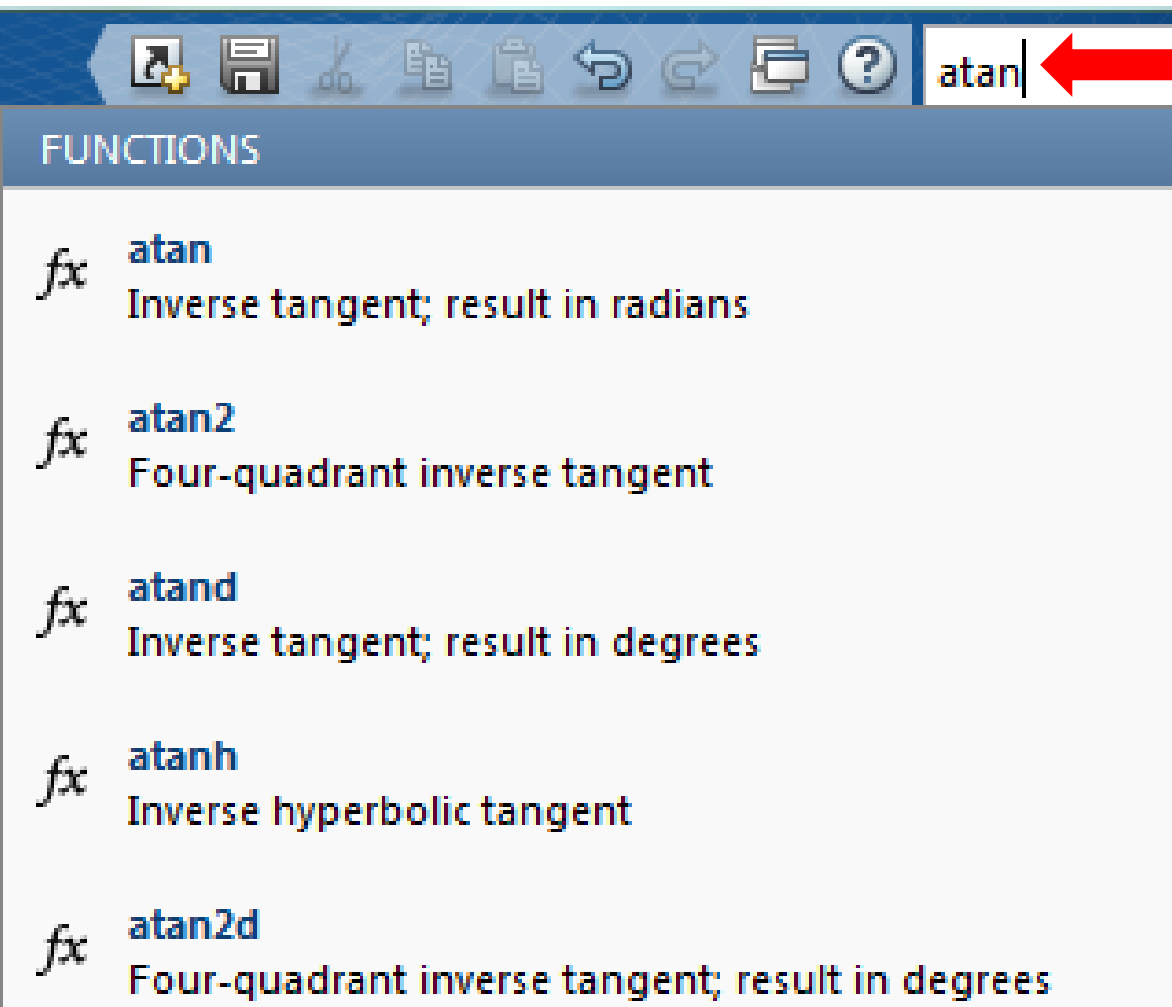
Command Window

```
u =
    32.1041

fx >>
```



To learn how to use the **atan** (arc tangent) command, use the **Search Documentation** window:



The screenshot shows the MATLAB Search Documentation window. At the top, there is a search bar with the text 'atan' and a red arrow pointing to it. Below the search bar, there is a list of search results under the heading 'FUNCTIONS'. The results are as follows:

Function	Description
<i>fx</i> <b>atan</b>	Inverse tangent; result in radians
<i>fx</i> <b>atan2</b>	Four-quadrant inverse tangent
<i>fx</i> <b>atand</b>	Inverse tangent; result in degrees
<i>fx</i> <b>atanh</b>	Inverse hyperbolic tangent
<i>fx</i> <b>atan2d</b>	Four-quadrant inverse tangent; result in degrees

## **atan**

Inverse tangent; result in radians

### **Syntax**

$Y = \text{atan}(X)$

### **Description**

$Y = \text{atan}(X)$  returns the inverse tan

*fx* **atan** - Inverse tangent; result in radians

This MATLAB function returns the inverse tangent (arctangent) for each element of X.

[MATLAB > Mathematics > Elementary Math > Trigonometry](#)

The **Order of Precedence** is a very important concept for properly performing calculations.

Precedence	Operation
First	Parentheses, evaluated starting with the innermost pair.
Second	Exponentiation, evaluated from left to right.
Third	Multiplication and division with equal precedence, evaluated from left to right.
Fourth	Addition and subtraction with equal precedence, evaluated from left to right.

Using Order of Precedence principles, create a MATLAB script file to calculate Problem 1-15(a).

**15.** Use MATLAB to calculate

a.  $e^{(-2.1)^3} + 3.47 \log(14) + \sqrt[4]{287}$

Function	MATLAB syntax*
$e^x$	exp (x)
$\sqrt{x}$	sqrt (x)
$\ln x$	log (x)
$\log_{10} x$	log10 (x)
$\cos x$	cos (x)
$\sin x$	sin (x)
$\tan x$	tan (x)
$\cos^{-1} x$	acos (x)
$\sin^{-1} x$	asin (x)
$\tan^{-1} x$	atan (x)

**15.** Use MATLAB to calculate

a.  $e^{(-2.1)^3} + 3.47 \log(14) + \sqrt[4]{287}$

```
problem1_15a.m ×
1 -   clc
2 -   clear
3
4 -   u = exp((-2.1)^3) + 3.47*log10(14) + (287)^(1/4)
5
```

### Command Window

u =

8.0931

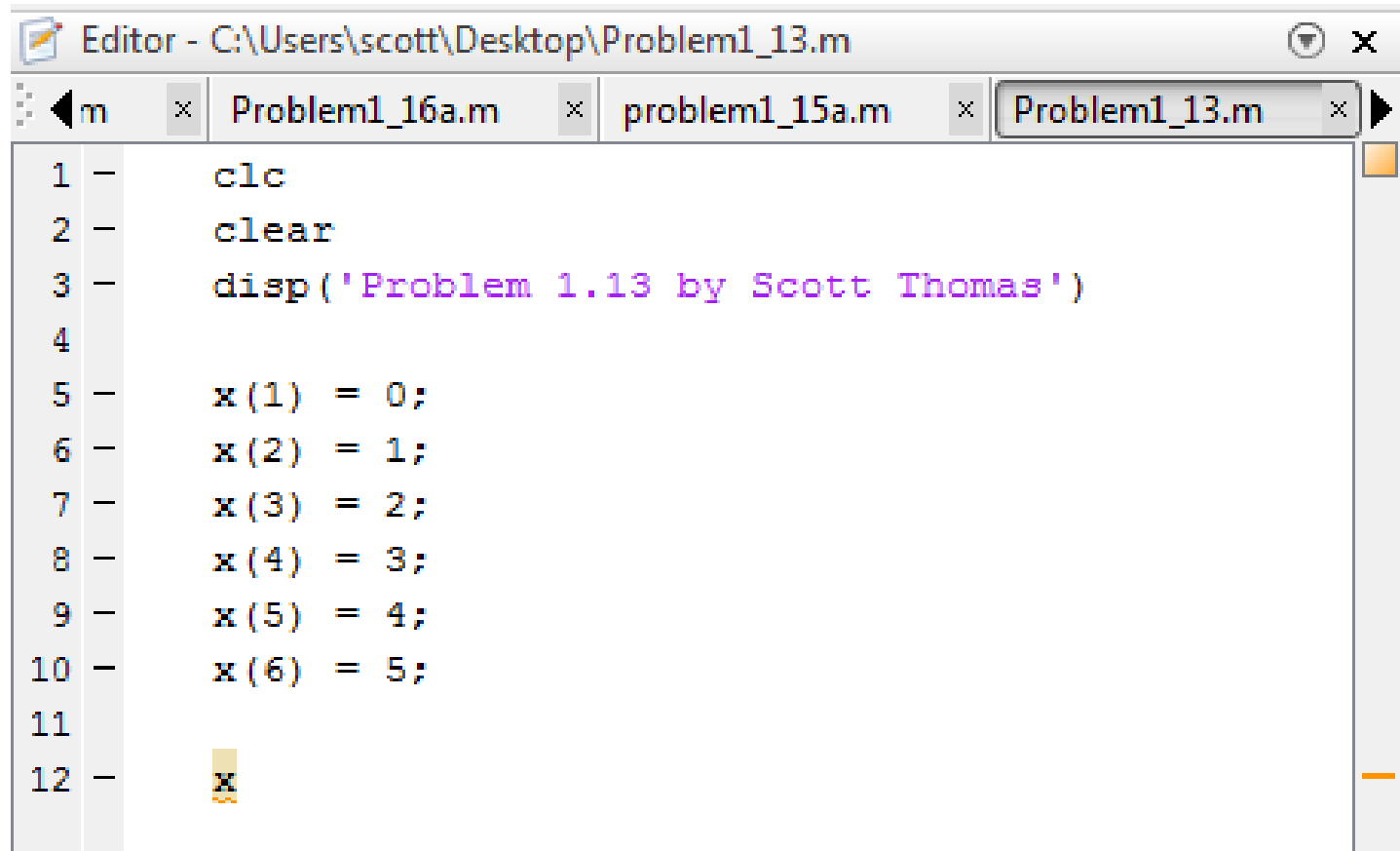
**15.** Use MATLAB to calculate

a.  $e^{(-2.1)^3} + 3.47 \log(14) + \sqrt[4]{287}$       b.  $(3.4)^7 \log(14) + \sqrt[4]{287}$

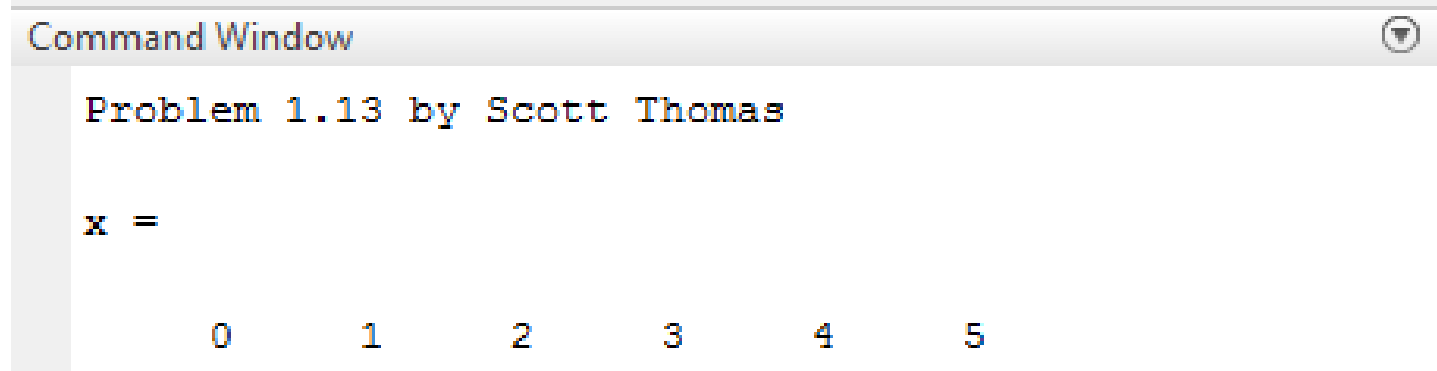
c.  $\cos^2\left(\frac{4.12\pi}{6}\right)$       d.  $\cos\left(\frac{4.12\pi}{6}\right)^2$

Function	MATLAB syntax*
$e^x$	exp (x)
$\sqrt{x}$	sqrt (x)
$\ln x$	log (x)
$\log_{10} x$	log10 (x)
$\cos x$	cos (x)
$\sin x$	sin (x)
$\tan x$	tan (x)
$\cos^{-1} x$	acos (x)
$\sin^{-1} x$	asin (x)
$\tan^{-1} x$	atan (x)

MATLAB can perform calculations on **Arrays** with the same ease as it does with single numbers (**Scalars**). Create and run a new **Script File** for Problem 1.13:



```
1 -   clc
2 -   clear
3 -   disp('Problem 1.13 by Scott Thomas')
4
5 -   x(1) = 0;
6 -   x(2) = 1;
7 -   x(3) = 2;
8 -   x(4) = 3;
9 -   x(5) = 4;
10 -  x(6) = 5;
11
12 -  x
```



```
Command Window

Problem 1.13 by Scott Thomas

x =

     0     1     2     3     4     5
```

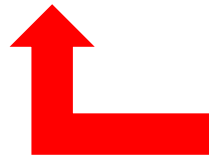
The new variable **x** is an **Array** of six separate numbers that can be acted on as a single unit. The syntax for an **Array** is

**Variable\_name(Index\_number) = Value**

The **Index Number** shows where the particular **Value** resides within the **Array**. Index numbers are integers starting with 1.

**x =**

0            1            2            3            4            5



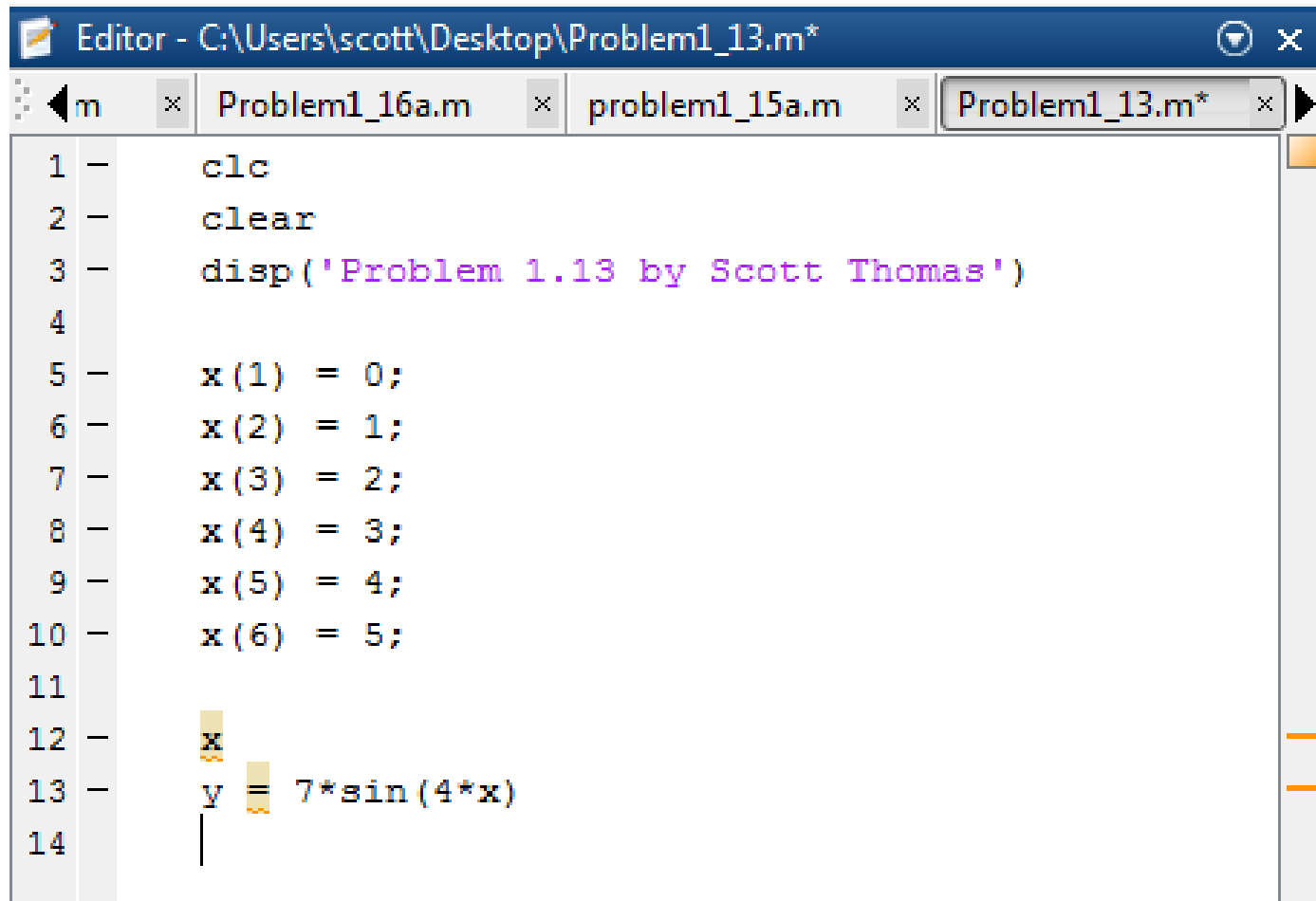
**Index Number = 1; Value = 0**

Notice that the **Workspace Window** shows all of the **Values** of variable **x**:

Workspace			
Name ▲	Value	Min	Max
x	[0,1,2,3,4,5]	0	5

Calculate the new variable **y (Array)** using the following equation in terms of variable **x**:

$$y = 7 \sin(4x)$$



The image shows a MATLAB Editor window titled "Editor - C:\Users\scott\Desktop\Problem1\_13.m\*". The window contains the following code:

```
1 -   clc
2 -   clear
3 -   disp('Problem 1.13 by Scott Thomas')
4
5 -   x(1) = 0;
6 -   x(2) = 1;
7 -   x(3) = 2;
8 -   x(4) = 3;
9 -   x(5) = 4;
10 -  x(6) = 5;
11
12 -  x
13 -  y = 7*sin(4*x)
14
```

Command Window

Problem 1.13 by Scott Thomas

x =

0 1 2 3 4 5

y =

Columns 1 through 5

0 -5.2976 6.9255 -3.7560 -2.0153

Column 6

6.3906

*fx* >>

Workspace

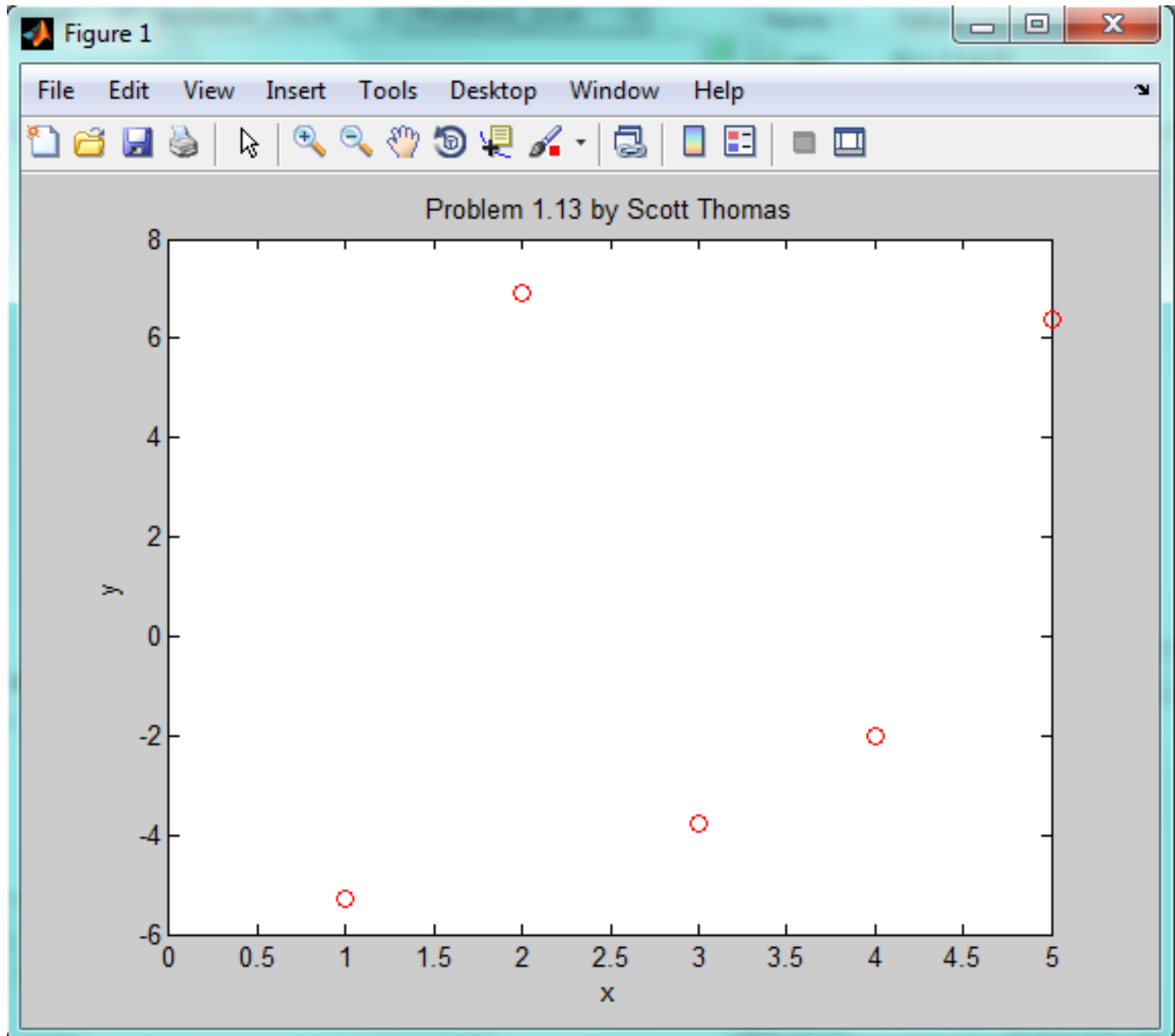
Name ▲	Value	Min	Max
x	[0,1,2,3,4,5]	0	5
y	[0,-5.2976,6.9255,-3.7560,-2.0153,6.3906]	-5.2976	6.9255



Plot the resulting **y Array** versus the **x Array** :

```
Editor - C:\Users\scott\Desktop\Problem1_13.m
Problem1_3a.m × Problem1_16a.m × problem1_15a.m × Problem1_13.m ×
1 -     clc
2 -     clear
3 -     disp('Problem 1.13 by Scott Thomas')
4
5 -     format short
6
7 -     x(1) = 0;
8 -     x(2) = 1;
9 -     x(3) = 2;
10 -    x(4) = 3;
11 -    x(5) = 4;
12 -    x(6) = 5;
13
14 -    x;
15 -    y = 7*sin(4*x);
16
17 -    figure
18 -    plot(x,y,'ro'), xlabel('x'),ylabel('y')
19 -    title('Problem 1.13 by Scott Thomas')
20 -    |
```

The Graphics Window appears, showing the graph of  $y$  versus  $x$ :



Create a new **Array x2**, use it to create a new **Array y2**, and plot **y2** versus **x2**:

```
1 -      clc
2 -      clear
3 -      disp('Problem 1.13 by Scott Thomas')
4
5 -      format short
6
7 -      x(1) = 0;
8 -      x(2) = 1;
9 -      x(3) = 2;
10 -     x(4) = 3;
11 -     x(5) = 4;
12 -     x(6) = 5;
13
14 -     x;
15 -     y = 7*sin(4*x);
16 -     x2 = 0:0.5:5;
17 -     y2 = 7*sin(4*x2);
18 -     figure
19 -     plot(x,y,'ro',x2,y2), xlabel('x'),ylabel('y')
20 -     title('Problem 1.13 by Scott Thomas')
```

The **Array x2** is created using an automated system, with syntax as follows:

**Variable\_Name = Start : Step : Stop**

where **Start** and **Stop** are the initial and final values (Range), and **Step** is the step size used to go between **Start** and **Stop**.

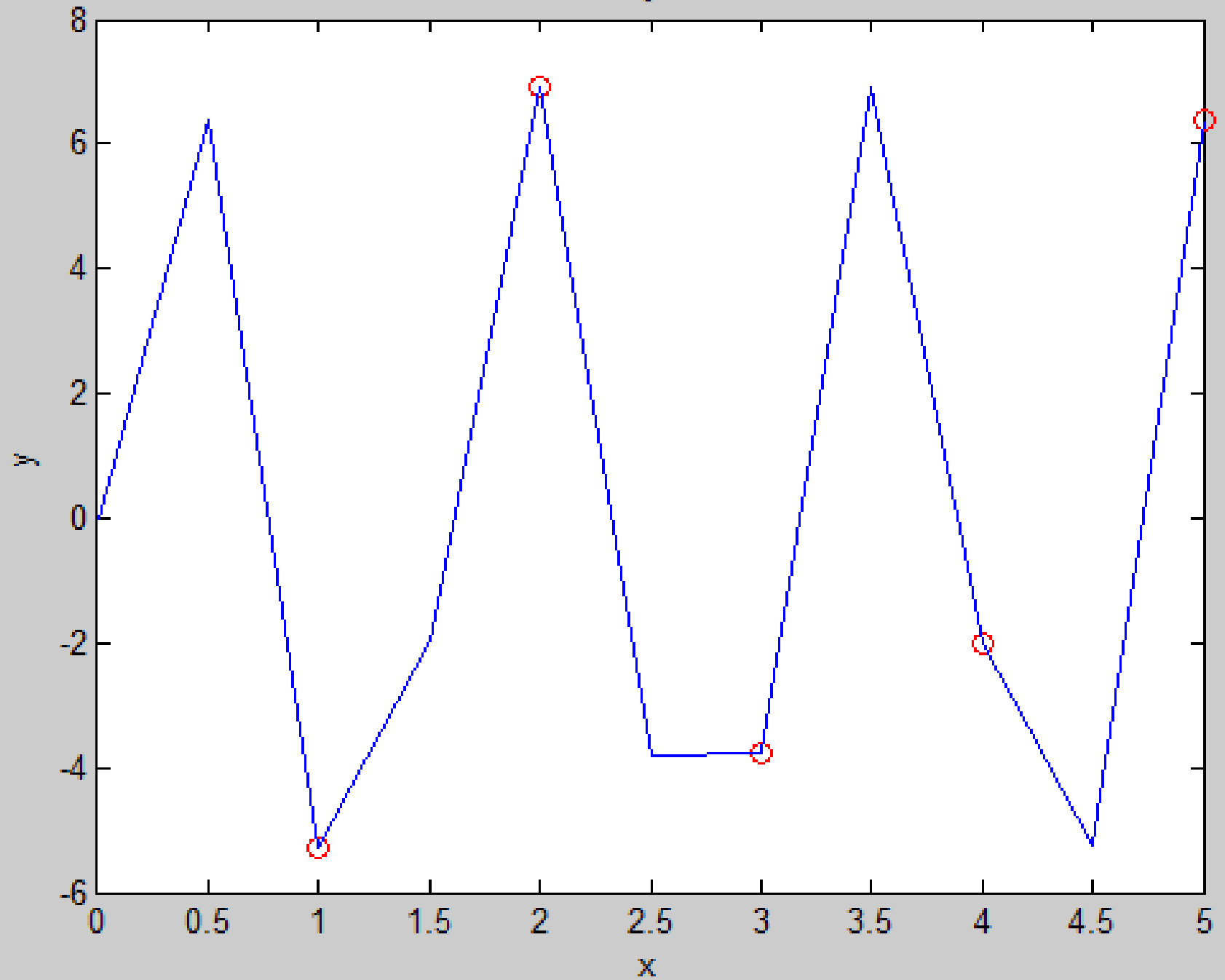
The Workspace Window now shows the new x2 and y2 variables as:

**<1x11 double>**

This means that they each have eleven numbers, arranged in one row and eleven columns:

Workspace				
Name ▲	Value	Min	Max	
ans	[0,1,2,3,4,5]	0	5	
x	[0,1,2,3,4,5]	0	5	
x2	<1x11 double>	0	5	
y	[0,-5.2976,6.9255,-3.7560,-2.0153,6.3906]	-5.2976	6.9255	
y2	<1x11 double>	-5.2976	6.9343	

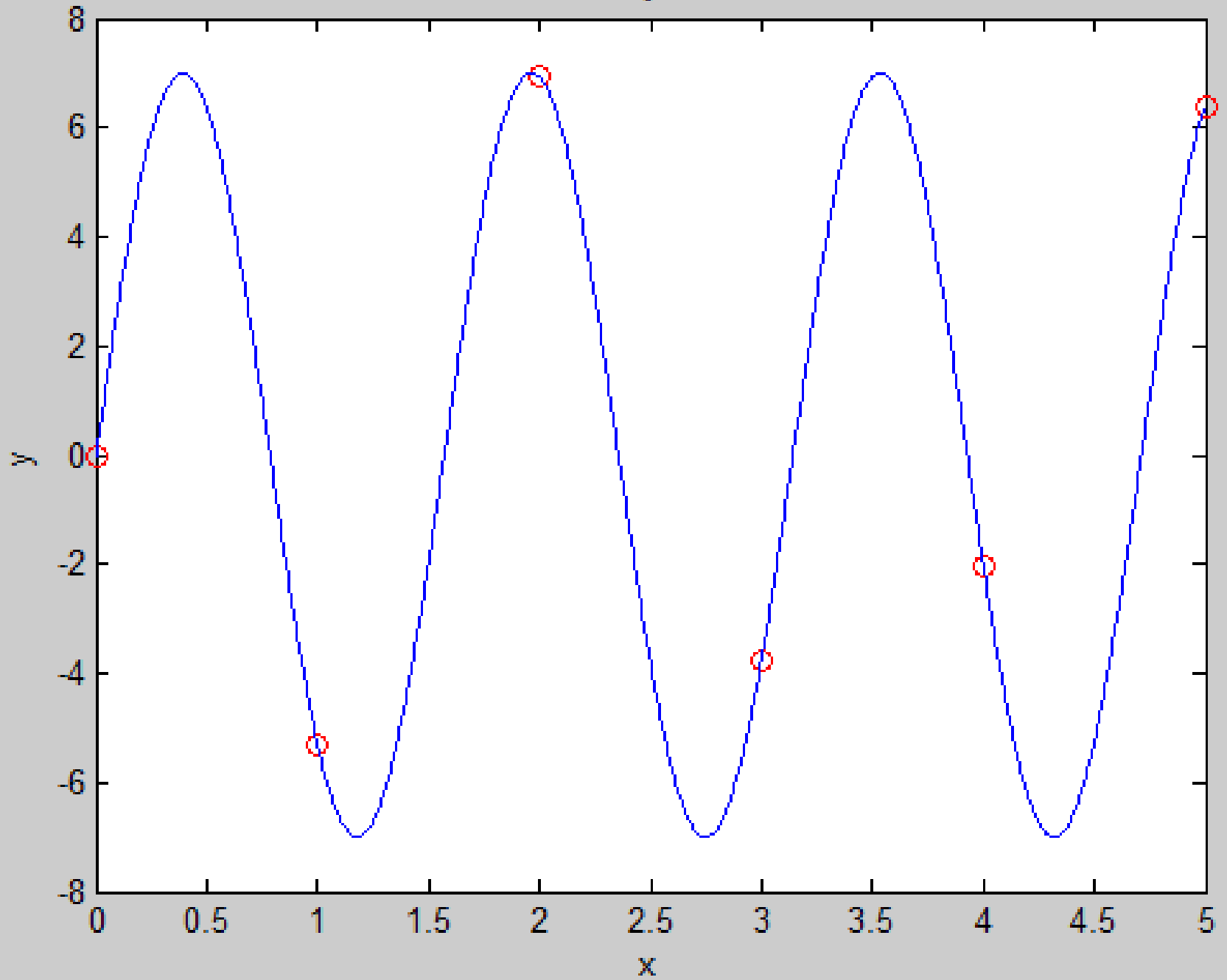
Problem 1.13 by Scott Thomas



Refine the plot by making the value for **Step** much smaller:

```
Editor - C:\Users\scott\Desktop\Problem1_13.m*
Problem1_3a.m × Problem1_16a.m × problem1_15a.m × Problem1_13.m* ×
1 -     clc
2 -     clear
3 -     disp('Problem 1.13 by Scott Thomas')
4
5 -     format short
6
7 -     x(1) = 0;
8 -     x(2) = 1;
9 -     x(3) = 2;
10 -    x(4) = 3;
11 -    x(5) = 4;
12 -    x(6) = 5;
13
14 -    x;
15 -    y = 7*sin(4*x);
16 -    x2 = 0:0.005:5;
17 -    y2 = 7*sin(4*x2);
18 -    figure
19 -    plot(x,y,'ro',x2,y2), xlabel('x'),ylabel('y')
20 -    title('Problem 1.13 by Scott Thomas')
21
```

Problem 1.13 by Scott Thomas



Notice that the number of values in **x2** and **y2** is now 1001.

Workspace			
Name ▲	Value	Min	Max
ans	[0,1,2,3,4,5]	0	5
x	[0,1,2,3,4,5]	0	5
x2	<1x1001 double>	0	5
y	[0,-5.2976,6.9255,-3.7560,-2.0153,6.3906]	-5.2976	6.9255
y2	<1x1001 double>	-7.0000	7.0000

What happens when you try to plot **y2** versus **x**? Try it!



```
Editor - C:\Users\scott\Desktop\Problem1_13.m
Problem1_3a.m x Problem1_16a.m x problem1_15a.m x Problem1_13.m x
13
14 - x;
15 - y = 7*sin(4*x);
16 - x2 = 0:0.005:5;
17 - y2 = 7*sin(4*x2);
18 - figure
19 - plot(x,y2)
20 % plot(x,y,'ro',x2,y2), xlabel('x'),ylabel('y')
21 - title('Problem 1.13 by Scott Thomas')
22
```

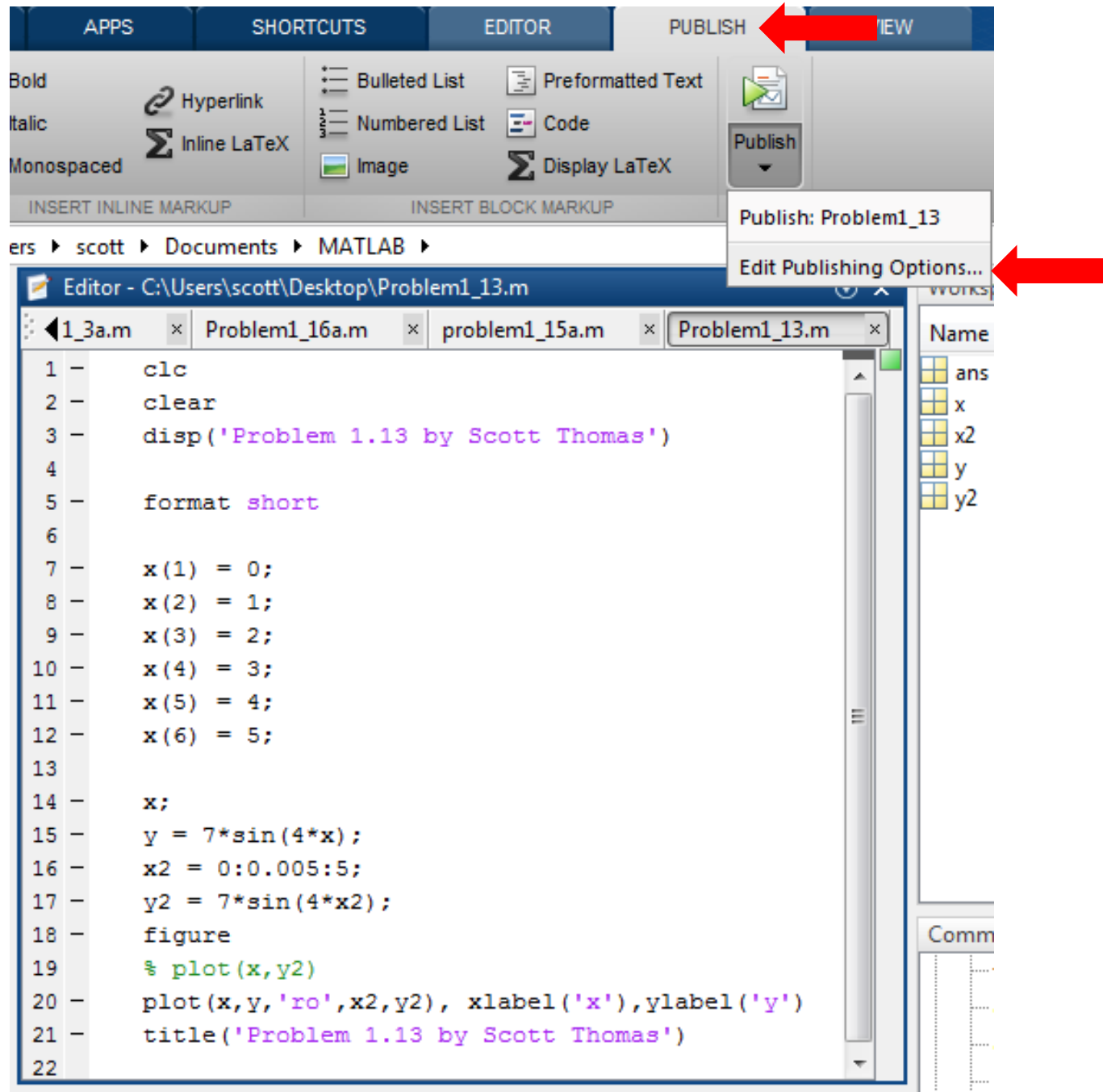
Command Window

Problem 1.13 by Scott Thomas  
Error using plot  
Vectors must be the same lengths.  
  
Error in Problem1\_13 (line 19)  
plot(x,y2)

Workspace

Name ▲	Value	Min	Max
ans	[0,1,2,3,4,5]	0	5
x	[0,1,2,3,4,5]	0	5
x2	<1x1001 double>	0	5
y	[0,-5.2976,6.9255,-3.7560,-2.0153,6.3906]	-5.2976	6.9255
y2	<1x1001 double>	-7.0000	7.0000

Use the **comment symbol (%)** to comment out the offensive command line. Go to the Publish tab as shown:



The screenshot displays the MATLAB software interface. At the top, the 'PUBLISH' tab is selected, indicated by a red arrow. The 'Publish' button in the toolbar is also highlighted with a red arrow. Below the toolbar, the file path 'ers > scott > Documents > MATLAB >' is visible. The main editor window shows a MATLAB script with the following code:

```
1 - clc
2 - clear
3 - disp('Problem 1.13 by Scott Thomas')
4
5 - format short
6
7 - x(1) = 0;
8 - x(2) = 1;
9 - x(3) = 2;
10 - x(4) = 3;
11 - x(5) = 4;
12 - x(6) = 5;
13
14 - x;
15 - y = 7*sin(4*x);
16 - x2 = 0:0.005:5;
17 - y2 = 7*sin(4*x2);
18 - figure
19 - % plot(x,y2)
20 - plot(x,y,'ro',x2,y2), xlabel('x'),ylabel('y')
21 - title('Problem 1.13 by Scott Thomas')
22
```

The right-hand side of the interface shows the 'Workspace' pane with variables: ans, x, x2, y, and y2. Below it, the 'Command Window' pane is partially visible.

# Change the Output File Format to PDF:

The screenshot shows the 'Edit Configurations' dialog box for a MATLAB configuration named 'Problem1\_13'. The 'Output settings' section is expanded, and the 'Output file format' dropdown menu is open, showing 'pdf' selected. A red arrow points to the 'pdf' option.

Problem1\_13.m

Problem1\_13.m

Problem1\_13

Publish configuration name: Problem1\_13

MATLAB expression:

```
% Modify expression to add input arguments.  
% Example:  
% a = [1 2 3; 4 5 6];  
% foo(a);  
  
Problem1_13
```

Publish settings: User Default Save As...

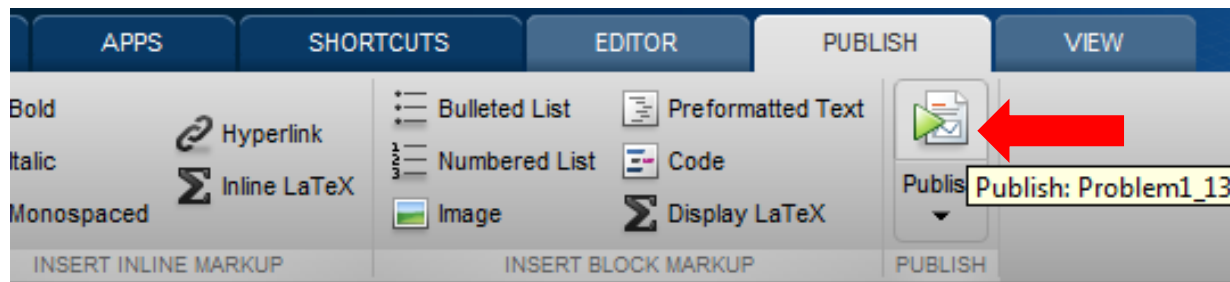
Output settings

Output file format	doc
Output folder	html
XSL file	xml
Figure settings	latex
Figure capture method	doc
Image Format	ppt
Use new figure	pdf
Max image width (pixels)	Inf
Max image height (pixels)	Inf
Create thumbnail	true
Code settings	
Include code	true

Select the output format for the published document.

Close Publish Help

Publish the file:



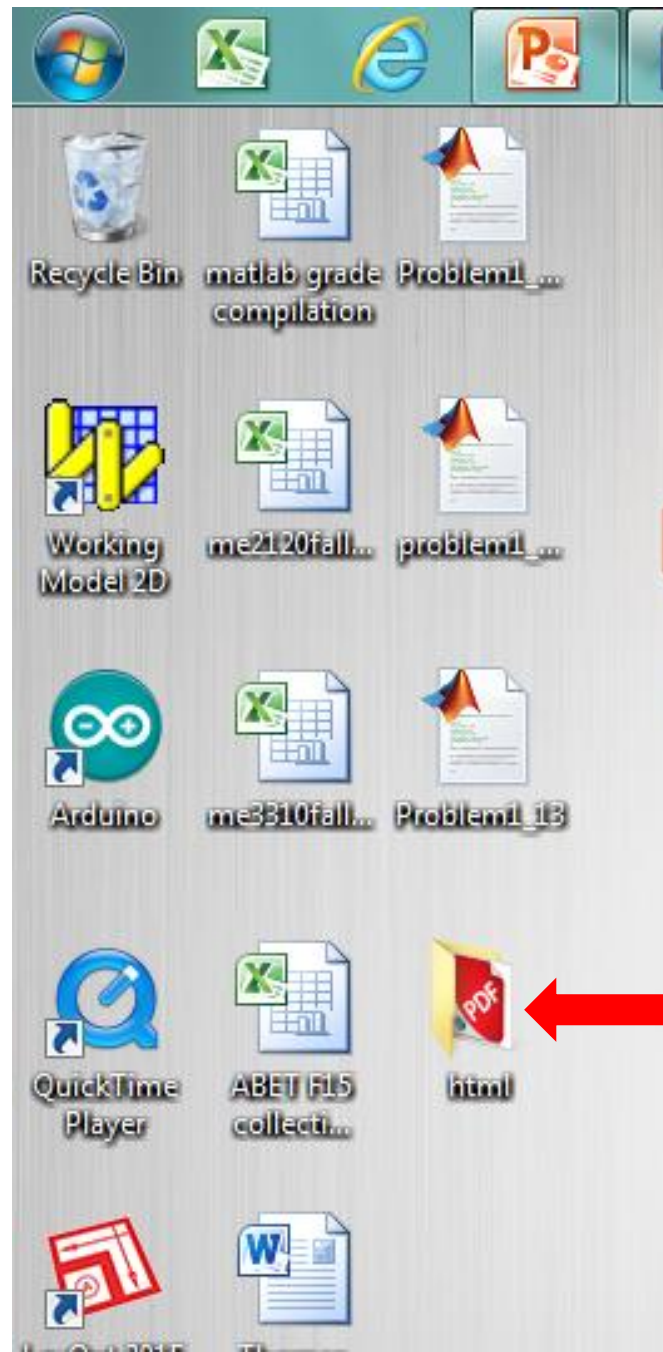
ers > scott > Documents > MATLAB >

The image shows the MATLAB Editor window with the file 'Problem1\_13.m' open. The code is as follows:

```
1 -   clc
2 -   clear
3 -   disp('Problem 1.13 by Scott Thomas')
4
5 -   format short
6
7 -   x(1) = 0;
8 -   x(2) = 1;
9 -   x(3) = 2;
10 -  x(4) = 3;
11 -  x(5) = 4;
12 -  x(6) = 5;
13
14 -  x;
15 -  y = 7*sin(4*x);
16 -  x2 = 0:0.005:5;
17 -  y2 = 7*sin(4*x2);
18 -  figure
19 -  % plot(x,y2)
20 -  plot(x,y,'ro',x2,y2), xlabel('x'),ylabel('y')
21 -  title('Problem 1.13 by Scott Thomas')
22
```

The workspace on the right shows variables: ans, x, x2, y, and y2. The Command Window is partially visible at the bottom right.

The file gets saved to a folder on the Desktop:



```
clc
clear
disp('Problem 1.13 by Scott Thomas')

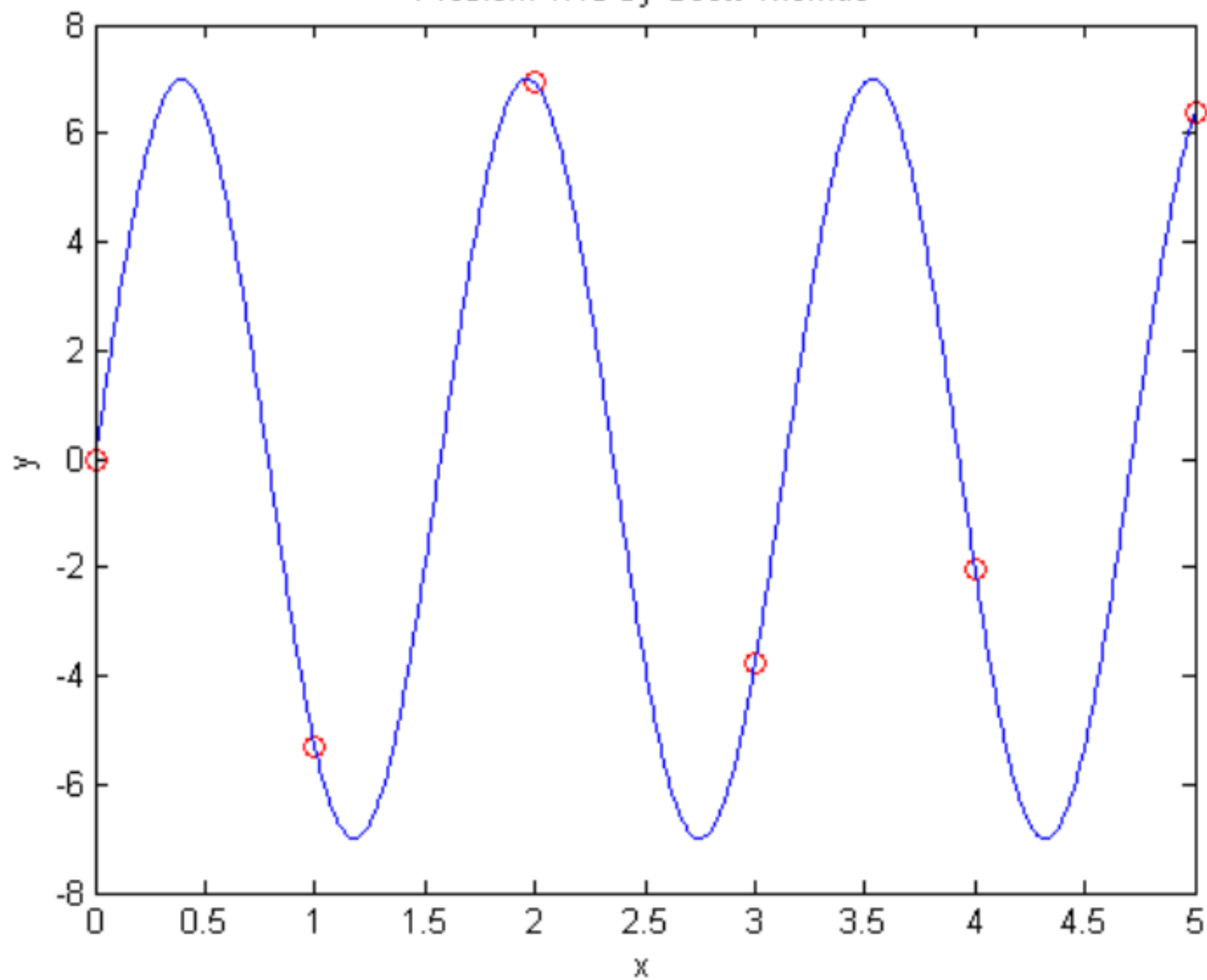
format short

x(1) = 0;
x(2) = 1;
x(3) = 2;
x(4) = 3;
x(5) = 4;
x(6) = 5;

x;
y = 7*sin(4*x);
x2 = 0:0.005:5;
y2 = 7*sin(4*x2);
figure
% plot(x,y2)
plot(x,y,'ro',x2,y2), xlabel('x'),ylabel('y')
title('Problem 1.13 by Scott Thomas')
```

*Problem 1.13 by Scott Thomas*

Problem 1.13 by Scott Thomas



Use MATLAB to calculate

$$a. \frac{3}{4} (6) (7^2) + \frac{4^5}{7^3 - 145}$$

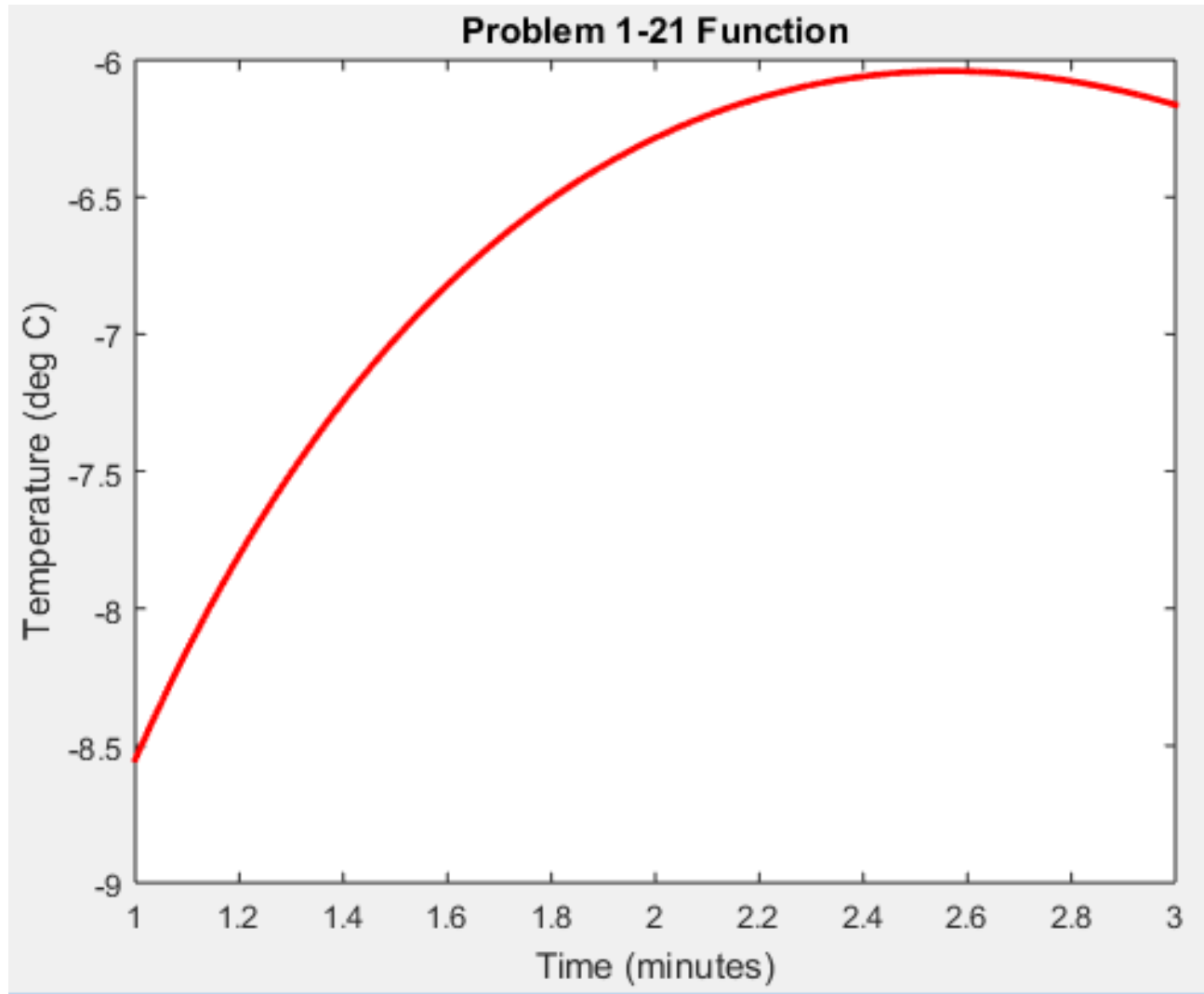
$$b. \frac{48.2(55) - 9^3}{53 + 14^2}$$

$$c. \frac{27^2}{4} + \frac{319^{4/5}}{5} + 60(14)^{-3}$$

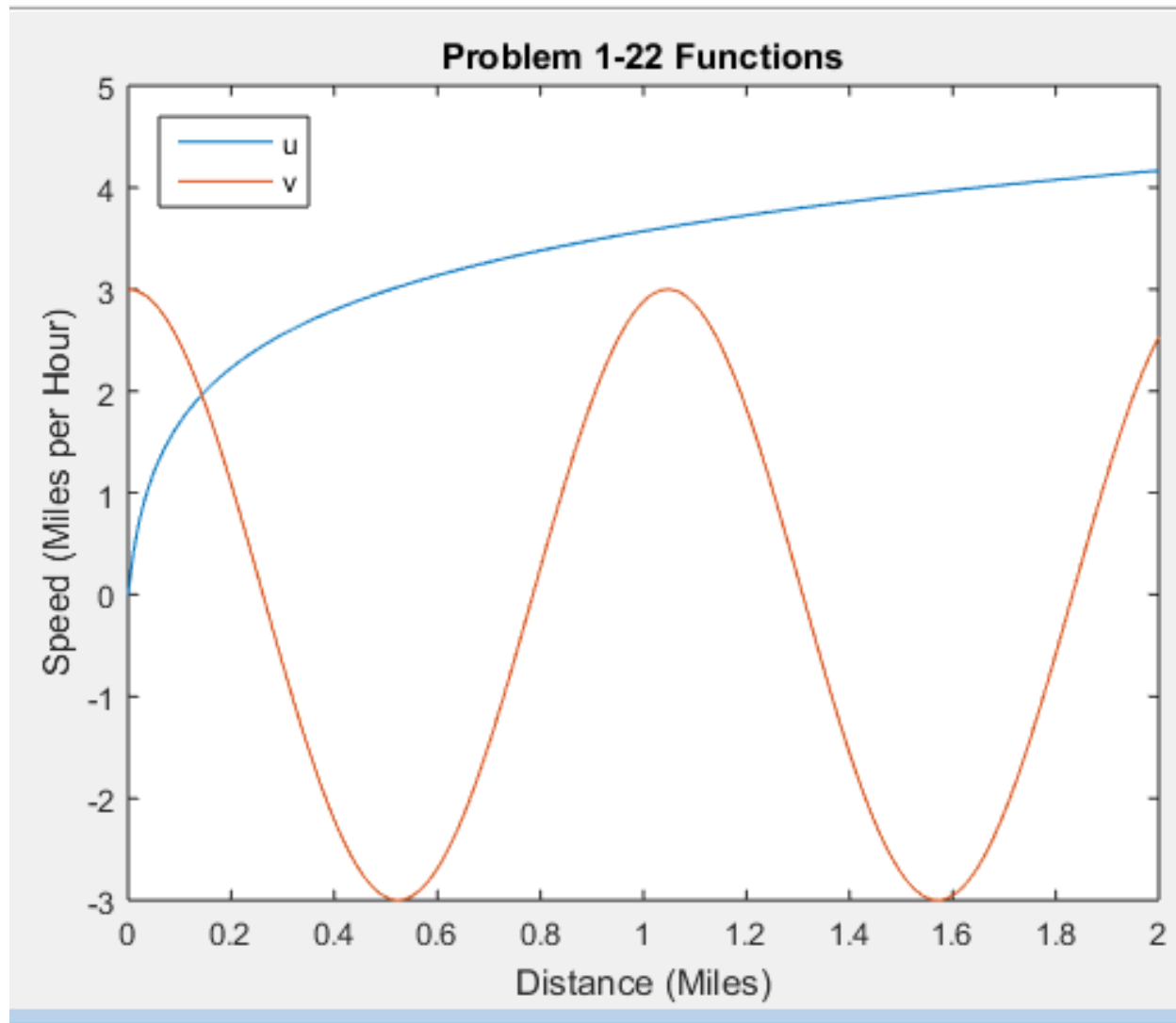


The volume of a sphere is given by  $V = 4\pi r^3/3$ , where  $r$  is the radius. Use MATLAB to compute the radius of a sphere having a volume 40 percent greater than that of a sphere of radius 4 ft.

Use MATLAB to plot the function  $T = 6 \ln t - 7e^{0.2t}$  over the interval  $1 \leq t \leq 3$ . Put a title on the plot and properly label the axes. The variable  $T$  represents temperature in degrees Celsius; the variable  $t$  represents time in minutes.



Use MATLAB to plot the functions  $u = 2 \log_{10}(60x + 1)$  and  $v = 3 \cos(6x)$  over the interval  $0 \leq x \leq 2$ . Properly label the plot and each curve. The variables  $u$  and  $v$  represent speed in miles per hour; the variable  $x$  represents distance in miles.



A *cycloid* is the curve described by a point  $P$  on the circumference of a circular wheel of radius  $r$  rolling along the  $x$  axis. The curve is described in parametric form by the equations

$$x = r (\phi - \sin \phi)$$

$$y = r (1 - \cos \phi)$$

Use these equations to plot the cycloid for  $r = 10$  in. and  $0 \leq \phi \leq 4\pi$ .

