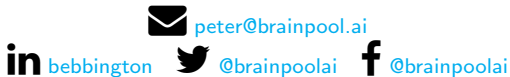


MATLAB for Finance Course: Session 1

Dr. Peter A. Bebbington

Brainpool AI



March 12, 2023

ABOUT ME

- 2009: MPhys. Physics From The University of Manchester.
- 2009-2012: Freelance Software Engineer.
- 2012: MSc. Complex System Modelling From King's College London.
- 2013: MRes. Financial Computing From University College London.
- 2017: PhD. Physics From University College London.
- 2015-2017: Research Associate at a Multi-Strategy Hedge Fund.
- 2017-Present: CTO for an A.I. Consultancy brainpool.ai
- 2021-Present: AI Director for daisy.ai

OBJECTIVE

- Matlab Desktop
- Command Window
- Arrays
- Matrices
- Set Functions
- System of Linear Equations
- Plotting

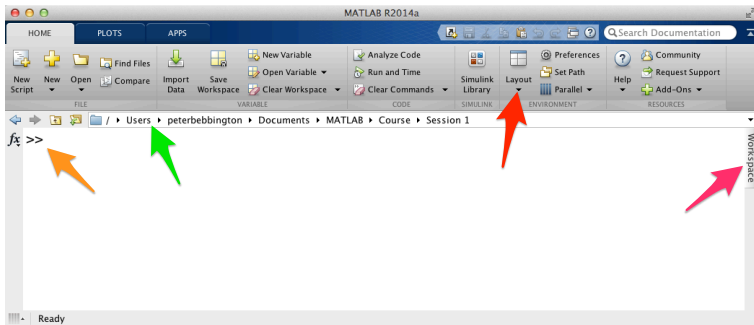
WHAT IS MATLAB?

- MATLAB is an interpreted programming language. The statements are translated in to machine code one by one in MATLAB's interpreter. In comparison, a compiled programming language like C has a program translated as a whole into machine code by the compiler.
- There are three ways in which you can work in MATLAB. From the Command prompt or MATLAB, writing scripts and writing functions.
- A script file can take no arguments, it is just a series of statements that will be executed sequentially. A function file can take and return arguments.

- The main MATLAB desktop has the following windows.
 - Command Window
 - Command History
 - Current Directory
 - Workspace
- MATLAB has extensive help files.
 - Text Based
 - Help Browser
 - Web Based: <http://www.mathworks.com/>

MATLAB LAYOUT

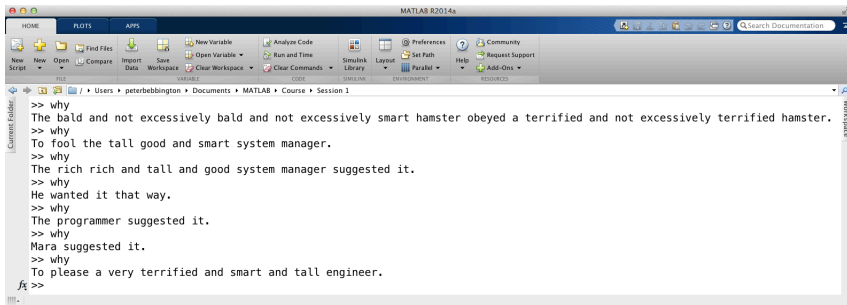
- Orange Arrow Command Line, Green Arrow Current Directory, Red Arrow Layout Editor and Pink Arrow Workspace.



- Makesure your current directory is something sensible for example folder on your desktop.

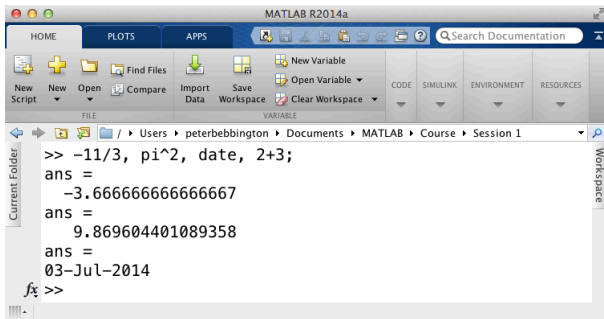
COMMAND WINDOW

- Lets look at typing and returning a simple command in to the command prompt.



NUMERICAL COMMANDS

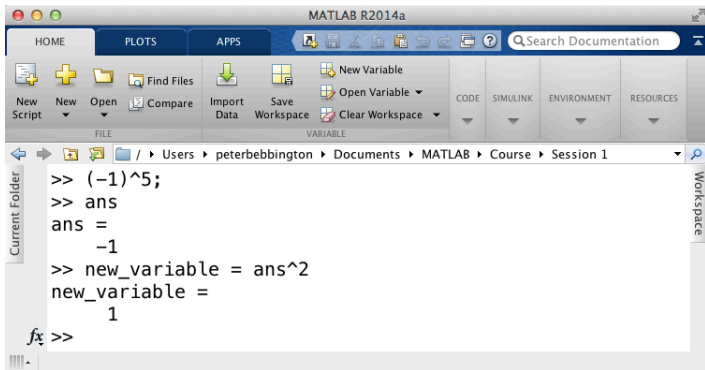
- Lets look at typing in some simple mathematical commands in the command prompt.



- For each command the output is created in the variable `ans` in the workspace.
- Notice that the semicolon stops the printing of the output to the command line and the comma allows the printing of the output to the command line.

VARIABLES

- The last answer can be called by returning `ans`.
- One can assign a variable with `=` operator.



The image shows the MATLAB R2014a software interface. The top menu bar includes HOME, PLOTS, and APPS. Below this is a toolbar with icons for New Script, New, Open, Find Files, Compare, Import Data, Save Workspace, New Variable, Open Variable, and Clear Workspace. The main workspace area displays the following MATLAB commands and their outputs:

```
>> (-1)^5;  
>> ans  
ans =  
    -1  
>> new_variable = ans^2  
new_variable =  
     1  
fx >>
```

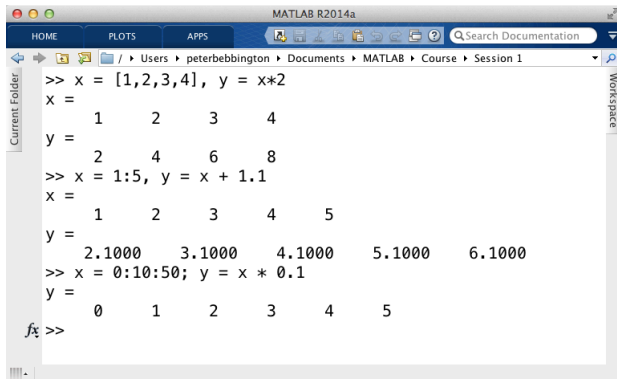
The left sidebar shows the Current Folder and the right sidebar shows the Workspace.

COMMANDS AND SHORTCUTS

Command	Meaning
<code>whos</code>	Gives the sizes and types of all loaded variables
<code>diary('filename')</code>	Save all commands in 'filename'
<code>diary on/off</code>	Set the mode of diary to on or off
<code>save 'workspace.mat'</code>	Save the current workspace in 'workspace.mat'
<code>load 'workspace.mat'</code>	Load 'workspace.mat' into the current workspace
<code>close all</code>	Closes all figure windows
<code>clear all</code>	Clears from memory all loaded variables
<code>clc</code>	Clear command windows
<code>ctrl+c</code>	Stops execution of a programme
<code>ctrl+] or [/ ⌘+] or [</code>	Indent the selected text forward or backward
<code>ctrl+R or T / ⌘+/ or T</code>	Comment or uncomment selected text
<code>↑ ↓</code>	Helps to search through old commands
<code>quit or exit</code>	Exit Matlab

ARRAYS

- Arrays are equivalent to vectors.
- Lets create an array of homogeneous variables (vectors) and operate on collectively on elements.



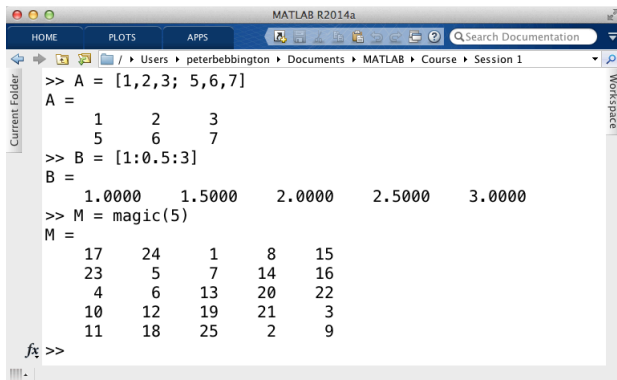
The image shows the MATLAB R2014a interface. The Command Window displays the following code and its output:

```
>> x = [1,2,3,4], y = x*2
x =
     1     2     3     4
y =
     2     4     6     8
>> x = 1:5, y = x + 1.1
x =
     1     2     3     4     5
y =
    2.1000    3.1000    4.1000    5.1000    6.1000
>> x = 0:10:50; y = x * 0.1
y =
     0     1     2     3     4     5
fx >>
```

The interface includes a top menu bar with 'HOME', 'PLOTS', and 'APPS'. Below it is a toolbar with various icons and a 'Search Documentation' field. The current directory is shown as '/ Users / peterbebbington / Documents / MATLAB / Course / Session 1'. The 'Current Folder' and 'Workspace' panels are visible on the left and right sides, respectively.

BUILDING MATRICES

- Matrix Constructor:
 - Column Separator ,
 - Row Separator ;
 - Uniformly Spaced :



The image shows a screenshot of the MATLAB R2014a software interface. The title bar indicates 'MATLAB R2014a'. The main window displays the Command Window with the following commands and outputs:

```
>> A = [1,2,3; 5,6,7]
A =
     1     2     3
     5     6     7

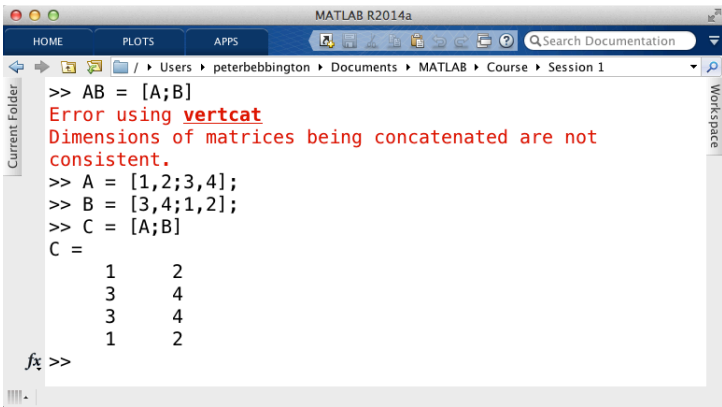
>> B = [1:0.5:3]
B =
    1.0000    1.5000    2.0000    2.5000    3.0000

>> M = magic(5)
M =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

The interface includes a 'Current Folder' pane on the left showing the path '/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1' and a 'Workspace' pane on the right. The top menu bar includes 'HOME', 'PLOTS', and 'APPS' tabs, along with a 'Search Documentation' bar.

CONCATENATING

- Matrices can be combined if and only if they have the correct dimensions.
- To do this one uses the `[]` operator.



The image shows a screenshot of the MATLAB R2014a software interface. The main window displays a command window with the following text:

```
>> AB = [A;B]
Error using vertcat
Dimensions of matrices being concatenated are not
consistent.
```

Below the error message, the user has defined matrices A, B, and C:

```
>> A = [1,2;3,4];
>> B = [3,4;1,2];
>> C = [A;B]
C =
     1     2
     3     4
     3     4
     1     2
```

The interface includes a top menu bar with 'HOME', 'PLOTS', and 'APPS' tabs. A search bar on the right says 'Search Documentation'. The current folder path is '/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1'. On the left, there is a 'Current Folder' pane, and on the right, a 'Workspace' pane.

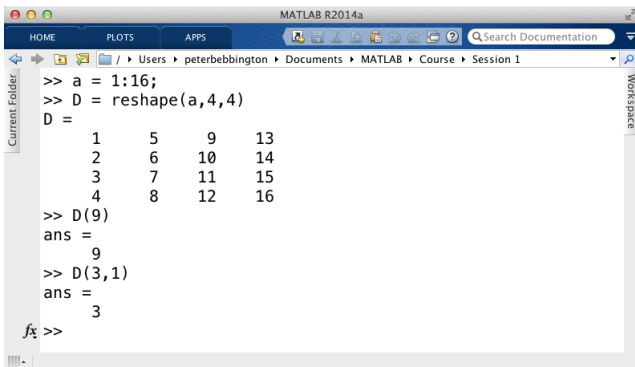
ADDRESSING MATRIX ELEMENTS

- Round Brackets are used to address Matrix elements: $D(r,c)$.
- It is also possible to address matrix elements via vectors:
 $D([r, c])$.

1,1	1,2	1,...	1, c_n
2,1	2,2	2,...	2, c_n
...,1	...,2	...,...	..., c_n
$r_m,1$	$r_m,2$	$r_m,...$	r_m,c_n

MATRIX ELEMENTS EXAMPLE

- Notice we have used the `reshape` function which can take an array as an argument and change the shape to another array with consistent dimensions.



The image shows a screenshot of the MATLAB R2014a command window. The window has a blue header bar with 'HOME', 'PLOTS', and 'APPS' tabs. Below the header is a search bar labeled 'Search Documentation'. The current directory path is displayed as '/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1'. The command window shows the following code and output:

```
>> a = 1:16;  
>> D = reshape(a,4,4)  
D =  
     1     5     9    13  
     2     6    10    14  
     3     7    11    15  
     4     8    12    16  
  
>> D(9)  
ans =  
     9  
  
>> D(3,1)  
ans =  
     3  
  
fx >>
```

MATRIX OPERATIONS

Syntax	Operation
'	Transpose
+	Addition
-	Subtraction
*	Matrix Multiplication
.*	Elementwise Multiplication
^	Matrix Power
.^	Elementwise Power

SPECIAL MATRICES

- It is convenient in MATLAB to be able to build special matrices without the necessity of long procedures. The reason will become more evident as we progress through the MATLAB course.

Command	Meaning
<code>ones(a,b)</code>	$a \times b$ matrix filled with ones
<code>zeros(a,b)</code>	$a \times b$ matrix filled with zeros
<code>eye(a)</code>	$a \times a$ identity matrix
<code>repmat(e,a,b)</code>	Replicates in $a \times b$ tiles the element e
<code>rand(a,b)</code>	$a \times b$ random matrix (uniformly distributed)
<code>randn(a,b)</code>	$a \times b$ random matrix (normally distributed)
<code>linspace(s,e,nr)</code>	Creates a linearly spaced row vector
<code>logspace(s,e,nr)</code>	Creates a log spaced row vector

SPECIAL MATRICES EXAMPLE

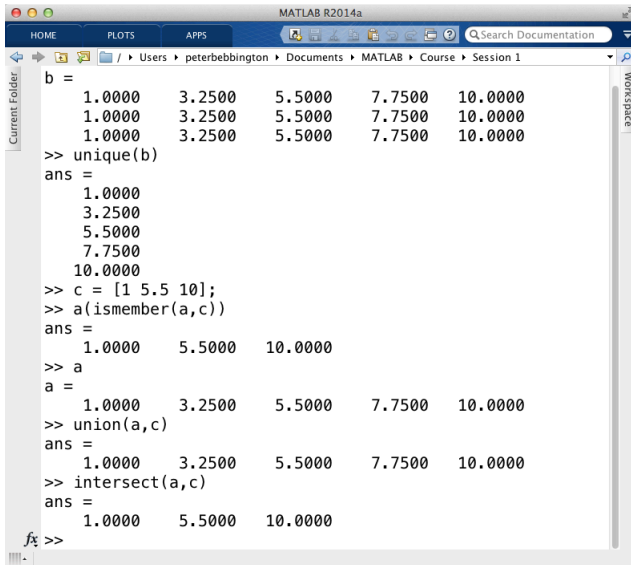
```
MATLAB R2014a
HOME PLOTS APPS Search Documentation
/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1

>> a = linspace(1,10,5)
a =
    1.0000    3.2500    5.5000    7.7500   10.0000
>> length(a), size(a)
ans =
     5
ans =
     1     5
>> b = repmat(a,3,1)
b =
    1.0000    3.2500    5.5000    7.7500   10.0000
    1.0000    3.2500    5.5000    7.7500   10.0000
    1.0000    3.2500    5.5000    7.7500   10.0000
fx >>
```

SET FUNCTIONS

Command	Meaning
<code>unique(a)</code>	Unique elements of a vector a
<code>union(a,b)</code>	Union of sets a and b
<code>intersect(a,b)</code>	Intersection of sets a and b
<code>ismember(a,b)</code>	Boolean for a elements in b
<code>setdiff(a,b)</code>	Elements in a that are not in b

SET FUNCTIONS EXAMPLE



The image shows a MATLAB R2014a window with the following content:

```
MATLAB R2014a
HOME PLOTS APPS Search Documentation
/Users/peterbebbington/Documents/MATLAB/Course/Session 1

b =
    1.0000    3.2500    5.5000    7.7500   10.0000
    1.0000    3.2500    5.5000    7.7500   10.0000
    1.0000    3.2500    5.5000    7.7500   10.0000

>> unique(b)
ans =
    1.0000
    3.2500
    5.5000
    7.7500
   10.0000

>> c = [1 5.5 10];
>> a(ismember(a,c))
ans =
    1.0000    5.5000   10.0000

>> a
a =
    1.0000    3.2500    5.5000    7.7500   10.0000

>> union(a,c)
ans =
    1.0000    3.2500    5.5000    7.7500   10.0000

>> intersect(a,c)
ans =
    1.0000    5.5000   10.0000

fx >>
```

SYSTEMS OF LINEAR EQUATIONS

- As MATLAB is an interpreted programming language it is intuitive to solve problems in Matrix form rather than using control flow with if/while/for loops. Consider the following example: system of equations.

$$2x + 2y - 2z = 10$$

$$6x + 4y + 4z = 2$$

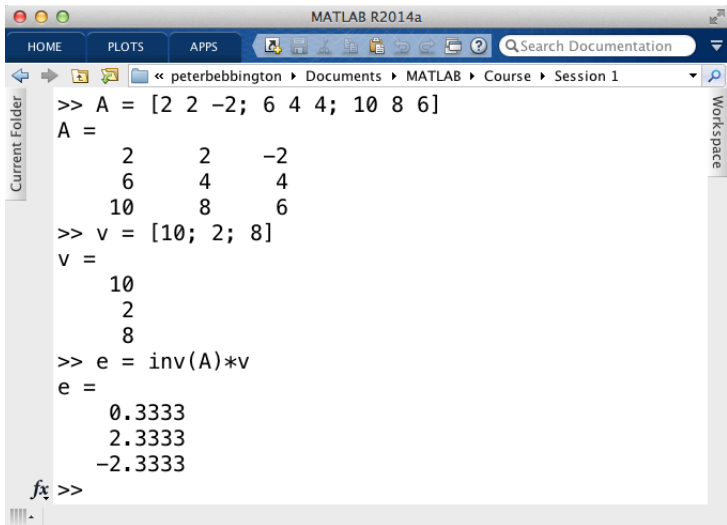
$$10x + 8y + 6z = 8$$

- These equations can be represented by

$$\bar{\bar{A}}\bar{e} = \bar{v}$$

SOLVING LINEAR EQUATIONS

- Solved by inverting matrix \bar{A} and multiplying it by \bar{v} .



The image shows a screenshot of the MATLAB R2014a software interface. The main window displays a script with the following commands and their outputs:

```
>> A = [2 2 -2; 6 4 4; 10 8 6]
A =
     2     2    -2
     6     4     4
    10     8     6

>> v = [10; 2; 8]
v =
    10
     2
     8

>> e = inv(A)*v
e =
    0.3333
    2.3333
   -2.3333
```

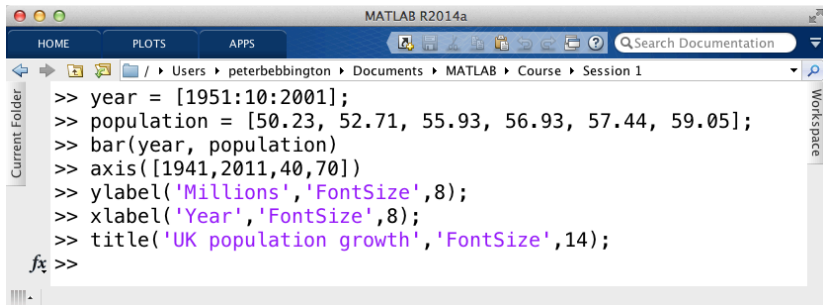
The interface includes a top toolbar with icons for Home, Plots, Apps, and a Search Documentation bar. The current folder is set to 'peterbebbington > Documents > MATLAB > Course > Session 1'. The workspace is visible on the right side of the window.

PLOTTING FUNCTIONS

Command	Meaning
<code>figure</code>	new figure window
<code>figure(n)</code>	n-th figure window
<code>title('string')</code>	add title to the window
<code>gtext('string')</code>	add text with the mouse
<code>legend('string')</code>	add legend
<code>grid on/off</code>	switch grid on/off
<code>subplot(m,n,p)</code>	multiple plots in one figure window
<code>hold on/off</code>	hold the current contents

- Note that once a figure is created Matlab can auto generate code to reproduce the plot by Figure -> File -> Generate Code

BAR PLOTS

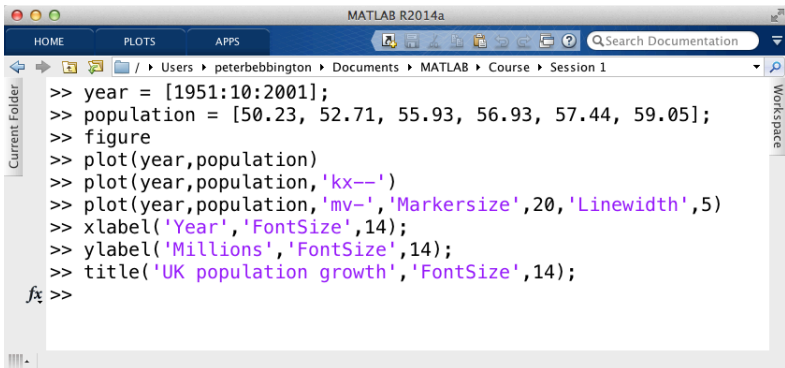


The image shows the MATLAB R2014a software interface. The top menu bar includes 'HOME', 'PLOTS', and 'APPS'. Below the menu bar is a toolbar with various icons and a 'Search Documentation' search bar. The main window displays a script editor with the following code:

```
>> year = [1951:10:2001];  
>> population = [50.23, 52.71, 55.93, 56.93, 57.44, 59.05];  
>> bar(year, population)  
>> axis([1941,2011,40,70])  
>> ylabel('Millions','FontSize',8);  
>> xlabel('Year','FontSize',8);  
>> title('UK population growth','FontSize',14);  
fx >>
```

The script defines a vector 'year' from 1951 to 2001 in increments of 10, and a vector 'population' with corresponding values. It then uses the 'bar' function to create a bar plot, sets the x-axis limits to [1941, 2011] and y-axis limits to [40, 70], and adds labels and a title with specified font sizes. The 'Current Folder' and 'Workspace' panels are visible on the left and right sides of the script editor.

LINE PLOTS

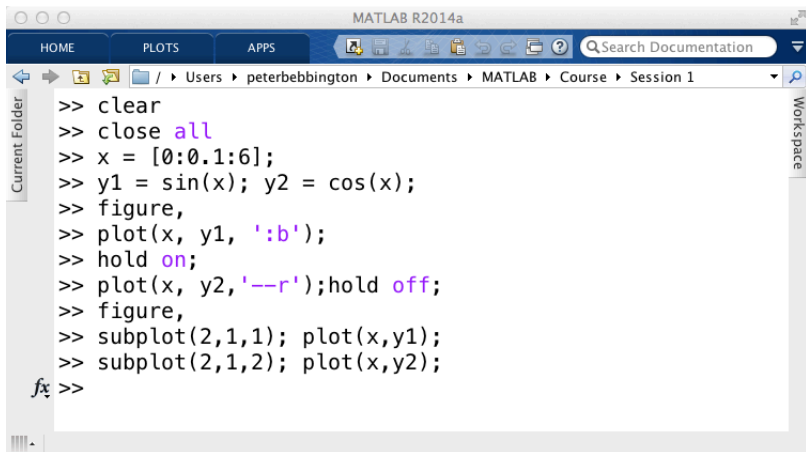


The image shows the MATLAB R2014a software interface. The top menu bar includes 'HOME', 'PLOTS', and 'APPS'. Below the menu bar is a toolbar with icons for file operations and a search bar labeled 'Search Documentation'. The current directory path is displayed as '/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1'. The main workspace area contains a script with the following MATLAB code:

```
>> year = [1951:10:2001];  
>> population = [50.23, 52.71, 55.93, 56.93, 57.44, 59.05];  
>> figure  
>> plot(year,population)  
>> plot(year,population,'kx--')  
>> plot(year,population,'mv-', 'Markersize',20, 'Linewidth',5)  
>> xlabel('Year','FontSize',14);  
>> ylabel('Millions','FontSize',14);  
>> title('UK population growth','FontSize',14);  
fx >>
```

The interface also features a 'Current Folder' pane on the left and a 'Workspace' pane on the right.

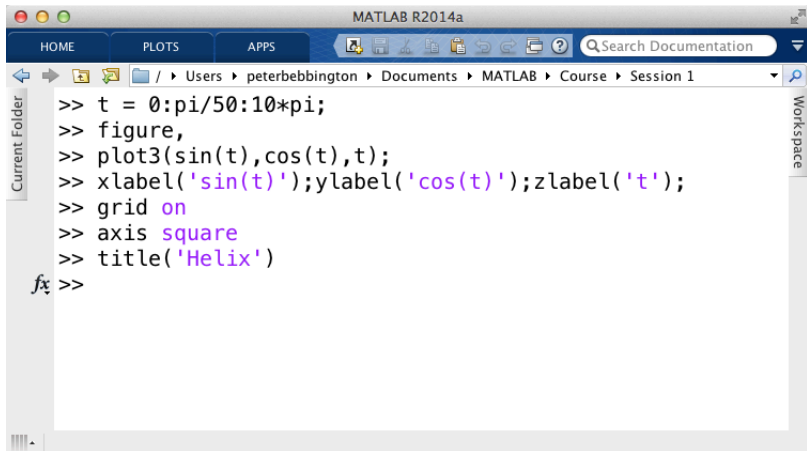
HOLD AND SUBPLOT



The image shows the MATLAB R2014a desktop environment. The top menu bar includes 'HOME', 'PLOTS', and 'APPS'. Below it is a toolbar with various icons and a 'Search Documentation' field. The current directory path is displayed as '/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1'. On the left, the 'Current Folder' pane is visible. On the right, the 'Workspace' pane is visible. The central command window contains the following MATLAB code:

```
>> clear
>> close all
>> x = [0:0.1:6];
>> y1 = sin(x); y2 = cos(x);
>> figure,
>> plot(x, y1, 'b');
>> hold on;
>> plot(x, y2, '--r'); hold off;
>> figure,
>> subplot(2,1,1); plot(x,y1);
>> subplot(2,1,2); plot(x,y2);
fx >>
```

3D PLOTS



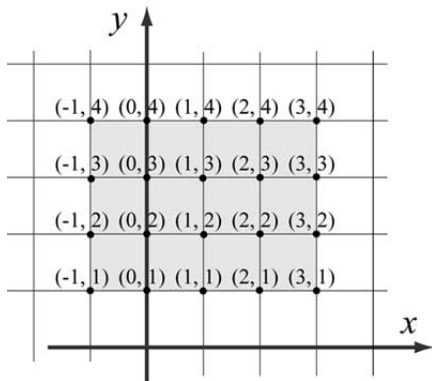
The image shows the MATLAB R2014a software interface. The top menu bar includes 'HOME', 'PLOTS', and 'APPS'. Below the menu bar is a search bar labeled 'Search Documentation'. The current directory path is displayed as '/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1'. The main workspace area contains a MATLAB script with the following code:

```
>> t = 0:pi/50:10*pi;  
>> figure,  
>> plot3(sin(t),cos(t),t);  
>> xlabel('sin(t)');ylabel('cos(t)');zlabel('t');  
>> grid on  
>> axis square  
>> title('Helix')  
fx >>
```

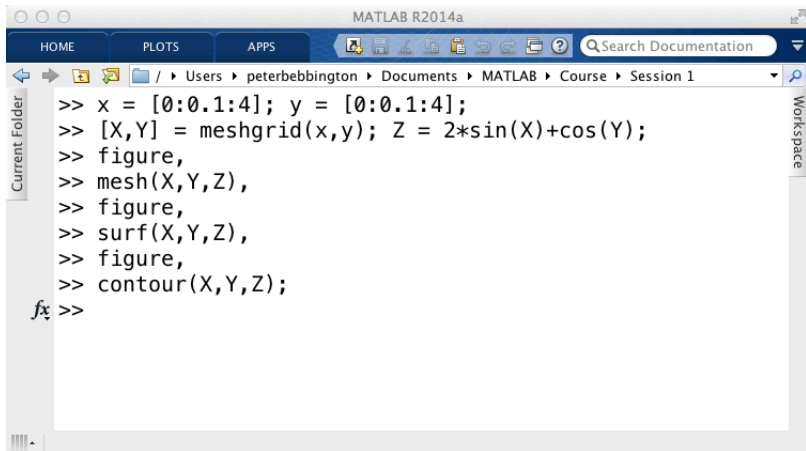
The left sidebar shows 'Current Folder' and the right sidebar shows 'Workspace'. The plot area is currently empty, but the code is designed to create a 3D plot of a helix.

MESH, SURFACE, CONTOUR

- Meshgrid creates the matrices of x and y co-ordinates
- mesh creates mesh plots
- surf creates surface plots
- contour creates contour plots.



MESH, SURFACE, CONTOUR PLOTS



The image shows the MATLAB R2014a software interface. The title bar at the top reads "MATLAB R2014a". Below it is a menu bar with "HOME", "PLOTS", and "APPS". To the right of the menu bar is a toolbar with icons for saving, opening, and other functions, followed by a search bar labeled "Search Documentation". Below the menu bar is a breadcrumb navigation path: "/ > Users > peterbebbington > Documents > MATLAB > Course > Session 1". The main workspace area contains a script with the following code:

```
>> x = [0:0.1:4]; y = [0:0.1:4];  
>> [X,Y] = meshgrid(x,y); Z = 2*sin(X)+cos(Y);  
>> figure,  
>> mesh(X,Y,Z),  
>> figure,  
>> surf(X,Y,Z),  
>> figure,  
>> contour(X,Y,Z);  
fx >>
```

On the left side of the workspace, there is a vertical label "Current Folder". On the right side, there is a vertical label "Workspace".