Stokes Modelling Workshop

14/06/2016

Introduction to Matlab

`www.maths.nuigalway.ie/ModellingWorkshop16/files`

MATLAB

As part of this crash course you will learn how to use MATLAB (for ”MATrix LABoratory”). MATLAB is an extraordinarily useful tool for all kinds of analysis and design.

Start the MATLAB program by clicking on the MATLAB symbol. The MATLAB command window will appear. You will see that the window has several panes. We will be entering commands into the ”Command Window”. Defined variables appear in the Workspace window (upper left side) and previous commands appear in the ”Command History” window (lower left).

It is possible to use the command window as a normal caculator. We can for instance write

```
>> 2+4
ans =
     6
```

```
>> 2^4
ans =
    16
```

Elementary functions

Type

```
>> help
```

in the command window and you will get a list useful functions, such as

```
sin, cos, tan, exp, abs, log, sqrt, round
```

It is also possible to write

```
>> help ops
>> doc ops
```

which gives you a list of useful operators, such as

```
+, -, *, ^, /, \, .*, ./
```

The most comprehensive help is given if you write `helpdesk`.

Special variables and constants

Some operators

`ans`	Most recent answer.	`Inf`	Infinity
`eps`	Floating-point relative accuracy.	`NaN`	Not a number
`i,j`	Imaginary unit.	`pi`	3.141592653589793

Time

`clock`	Wall clock.	`etime`	Elapsed time function.
`cputime`	Ellapse CPU time.	`tic, toc`	Stopwatch timer function
`date`	Calendar.		

Vectors

Create a row vector `v` with elements $1, 3, 4, 6$

```
>> v = [1,3,4,6]
v =
     1     3     4     6
```

and create a column vector `v` with elements $1, 3, 4, 6$

```
>> v1 = [1;3;4;6]
v1 =
      1
      3
      4
      6
```

```
>> w = v’
w =
     1
     3
     4
     6
```

Notitce that the *transpose* operator is thus given by `’`.

Vectors

Create a row vector v with elements $1, 2, \ldots, 7$

```
>> v = 1:7
v =
     1     2     3     4     5     6     7
```

and create a row vector v with elements $1.1, 1.3, \ldots, 1.9$ write

```
>> v = 1.1:0.2:1.9
v =
    1.1000    1.3000    1.5000    1.7000    1.9000
```

Matrices

To create a 2×3 matrix `M` with elements $1, 3, 4, 6, 7, 9$ write either

```
>> M = [1,3,4;6,7,9]
M =
     1     3     4
     6     7     9
```

or

```
>> M = [1 3 4
        6 7 9]
M =
     1     3     4
     6     7     9
```

or

Matrices

```
>> M(1,1) = 1; M(1,2) = 3; M(1,3) = 4;
   M(2,1) = 6; M(2,2) = 7; M(2,3) = 9;
>> M
M =
     1     3     4
     6     7     9
```

or

```
>> M(1,:) = [1,3,4]; M(2,:) = [6,7,9];
>> M
M =
     1     3     4
     6     7     9
```

or

Matrices

```
>> M(:,1) = [1;6]; M(:,2) = [3;7]; M(:,3) = [4;9];
>> M
M =
     1     3     4
     6     7     9
```

To create a 3×3 identity matrix `Id` write

```
>> Id = eye(3)
Id =
     1     0     0
     0     1     0
     0     0     1
```

Boolean operators and functions

Some operators

<	less than	>	greater than
<=	less than or equal to	>=	greater than or equal to
&	logical AND	|	logical OR
~	logical NOT	xor	logical EXCLUSIVE OR

Functions

all	True if all elements of a vector is true.
find	Find indicies of nonzero elements.

M-File scripts

When using MATLAB as mathematical computing tool, and not only a calculator, it is more convenient to write your code in a file that can be saved, instead of in the command window. By doing this you can return to the code at a later time. The extension of executable files in matlab is `.m`.

Scripts are the simplest kind of M-file because they have no input or output arguments. They are useful for automating series of MATLAB commands, such as computations that you have to perform repeatedly from the command line.

To open a new file chose File → New → M-File from the Menu bar and start typing your MATLAB code. As always: **Don't forget to save!!**

if...else...end

Consider the following piece of code

```
v = [4,6];
if v(1) > v(2),
   largest_value = v(1);
else
   largest_value = v(2);
end
```

What does the code do if `v = [4,4]` instead?

for...end

Consider the code

```
v = 1:2:9;
for k = 1:length(v),
   s(k) = v(k)^2;
end
```

This gives us the vector of squares of $1, 3, 5, 7, 9$

```
>> s
s =
     1     9    25    49    81
```

There is another way to create this vector

```
>> s = v.^2
s =
     1     9    25    49    81
```

plot

Plotting graphs is very important for anybody working in any applied science. MATLAB is very good for this purpose.

The function `PLOT(X,Y,S)` plots vector Y versus vector X, where S is a character string describing the way in which the data is shown. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up. If X is a scalar and Y is a vector, disconnected line objects are created and plotted as discrete points vertically at X.

To find lot of information on how to plot your data type `help plot` or `doc plot`.

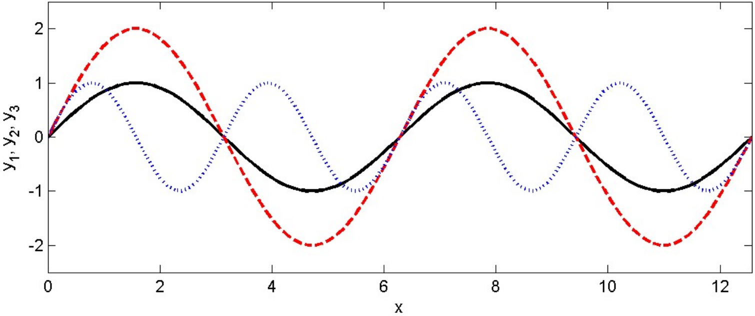
plot

Consider the following

```
x = 0:4*pi/100:4*pi;
y1 = sin(x);
y2 = 2*sin(x);
y3 = sin(2*x);
figure(7),hold on
plot(x,y1,'k','LineWidth',2)
plot(x,y2,'r--','LineWidth',2)
plot(x,y3,'b:','LineWidth',2)
xlabel('x'),ylabel('y_1, y_2, y_3')
axis([0 4*pi -2.5 2.5])
box on
hold off
```

plot

This code gives the following figure



2
1
0
-1
-2
y_1, y_2, y_3
0
2
4
6
8
10
12
x