Introduction to MATLAB 2

m-files

are text files with extension '.m' that contain the source code.
Matlab has its own editor.
To run the code in an m-file make sure the path is set to your work folder and type the name of the file without the extension in the matlab command window OR click the 'run' icon in the editor window.

Terminating lines

All code that works in the command window can be placed in an m-file to make programs. The output produced during the execution of the lines of code can be muted by putting semicolons at the end of the lines.

Pausing execution

The command called **pause** can be used to stop execution until the user presses a key. Execution can be delayed with the same command if an argument is provided. Alternatively you can put breakpoints by clicking in the left margin.

if Statements

The if statement controls the flow of the execution based on the result of the logical expression. For example in the following:

```
if logical_expression
        statements_1
end
```

statements_1 will be executed if the logical_expression evaluates to true.

if - else Statements

```
if logical_expression
        statements_1
else
        statements_2
end
```

statements_1 will be executed if the logical_expression evaluates to true. Otherwise, statements_2 will be executed.

e.g. The following will add 5 to x if the value of x is greater than 40. If the value of x is less than or equal to 40 it will subtract 5 from x.

```
if  x > 40
        x = x +5;
else
        x = x - 5;
end
```

The following will display the text 'OK' if x is greater than 10 and y is less than or equal to 2. If not, it will display 'FALSE'. disp is used to display the entity in parentheses.

```
if  x > 10 && y <= 2
        disp('OK');
else
        disp('FALSE');
end
```

for loops

```
a = 0;                          % sum of numbers from 1 to 10
for i = 1 : 10                  % same as sum( 1:10 )
    a = a + i;
end

b = 1;                          % product of odd numbers between 1 and 9
for i = 1: 2 :10                % same as prod( 1: 2: 10 )
        b = b * i;
end

c = 1;                          % product of numbers divisible by 3
for i = 3: 3: 20                % between 3 and 9
   if i > 10                    % same as prod( 3:3:9 )
        break;                  % breaks out of the loop
   end
   c = c * i;
end

c = 1;                          % product of numbers divisible by 3
for i = 3:9                     % between 3 and 9
   if mod(i, 3) ~= 0            % same as prod( 3:3:9 )
        continue;               % continues without executing the rest of the loop
   end
   c = c * i;
end
```

while loops

```
k = 0;                          % display numbers in increments of 0.5
while k < 12                    % similar to 0:0.5:11.5
    disp( k );
    k = k + 0.5;
end
```

switch statements

```
num = input('Enter a  number (one of -1, 0, 1): ');
switch num
   case -1
      disp('negative one')
   case 0
      disp('zero')
   case 1
      disp('positive one')
   otherwise
      disp('invalid input')
end
```

Functions

are declared in files which have the same names of the functions

Declaration in the file called multiply.m

```
function c = multiply(a, b)
    c = a * b;
```

Call from either the command line or another script, or function.

```
ret_value =  multiply(3, 4);
```

Notes on functions:

- Functions have their own local scope. Variables created within a function are not accessible from other code outside of the function.
- If you are writing a regular matlab script in an m-file and not using functions, put the **clear** command in the first line to remove all variables in the workspace every time the script is run.
- You can declare a single function in an m-file and call it from other scripts using the same name.
- You can declare more than one function in an m-file.

Graphically displaying Array Contents

Plot can be used to visualize sequences but is not useful for arrays. The contents of a numeric array can be mapped to colors by using **imagesc**. This function will automatically map the range of numbers in the array to the colormap in use. The defaut colormap is 'parula':

Display functions like **plot** and **imagesc** will draw on the current figure. Multiple figures (windows) can be opened by using the command **figure**. The current figure can be selected by the same command. E.g. figure(1); imagesc(A); figure(2); imagesc(B);