

Matlab Programming Arrays and Scripts ¹ ²

Mili I. Shah

September 10, 2009

¹*Matlab, An Introduction with Applications, 2nd ed.* by Amos Gilat

²*Matlab Guide, 2nd ed.* by D. J. Higham and N. J. Higham

Matrix Operations

- Matrix operations can be performed by using (+,-,*,/,^)

```
>> A = [1 4 2; 1 6 3; 1 3 2]
```

```
A =
```

```
1 4 2
```

```
1 6 3
```

```
1 3 2
```

```
>> B = [1 2; 1 5; 3 6]
```

```
B =
```

```
1 2
```

```
1 5
```

```
3 6
```

```
>> A*B
```

```
ans =
```

```
11 34
```

```
16 50
```

```
10 29
```

```
>> B*A?
```

Systems of Equations

- Can solve systems of equations with `inv` or the backslash (`\`)

```
>> A = [1 4 2; 1 6 3; 1 3 2]
```

```
A =
```

```
1 4 2
```

```
1 6 3
```

```
1 3 2
```

```
>> b = [1; 1; 3]
```

```
b =
```

```
1
```

```
1
```

```
3
```

```
>> inv(A)*b;
```

```
>> A\b;
```

- See the answer for both is $[1, -2, 4]^T$

Element-wise operations

- Element-wise matrix operations can be performed using $(.*, ./, .^)$

```
>> A = [1 4 3; 2 5 6]
```

```
A =
```

```
1 4 3
```

```
2 5 6
```

```
>> B = [4 6 3; 7 3 5]
```

```
B =
```

```
4 6 3
```

```
7 3 5
```

```
>> A.*B
```

```
ans =
```

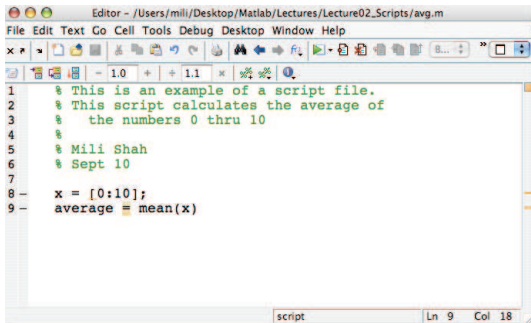
```
4 24 9
```

```
14 15 30
```

- Can create random matrices using the following commands
 - `rand`: Generates a single random number between 0 and 1
 - `rand(1,n)`: Generates an n element row vector of random numbers between 0 and 1
 - `rand(n)`: Generates an $n \times n$ matrix with random numbers between 0 and 1
 - `rand(m,n)`: Generates an $m \times n$ matrix with random numbers between 0 and 1
 - `randn`: Generates normally distributed numbers with mean 0 and standard deviation 1 (inputs same as `rand`)
 - `randperm(n)`: Generates a row vector with n elements that are random perturbations of integers 1 through n

Scripts

- A **script file** is a sequence of commands also called a program.
- The output is displayed in the Command Window
- A script file is convenient because it can be edited and executed many times
- Script files can be typed and edited in any text editor and then pasted into the Matlab editor
- Also called an m-file because the extension `.m` is used



The screenshot shows a MATLAB script editor window titled "Editor - /Users/mili/Desktop/Matlab/Lectures/Lecture02_Scripts/avg.m". The window contains the following script code:

```
1 % This is an example of a script file.
2 % This script calculates the average of
3 % the numbers 0 thru 10
4 %
5 % Mili Shah
6 % Sept 10
7
8 x = [0:10];
9 average = mean(x)
```

The status bar at the bottom of the window indicates "script" and "Ln 9 Col 18".

Script Output: disp

`disp`: Displays output on the screen

`disp(name of variable)` or `disp('text as string')`

```
>> A = [1 4 3; 2 5 6];  
>> disp(A)  
1 4 3  
2 5 6  
>> disp('Matlab is GREAT');  
Matlab is GREAT
```

- Note: `disp('')` displays an empty line
- Note: If printing a table, spaces may have to be added for tables to line up

Script Output: fprintf in Command Window

`fprintf`: Displays output on the screen or saves output to file

```
fprintf('text typed in as a string')
```

```
>> fprintf('Matlab is GREAT');  
Matlab is GREAT>>
```

- Note: New line is not created. Use `\n` for a new line
- Note: Use `\t` for a horizontal tab
- Can be used to display a mix of text and numerical data

```
fprintf('text %-5.2f additional text %-5.2f', var1,var2)
```

- Formatting numerical data:

`%Flag Precision(1st).Precision(2nd) Conversion`

- Flag: (-) Left justifies, (+) Prints sign, (0) Adds zeros if short
- Precision: (1st) is field width, (2nd) is # of digits rt of decimal
- Conversion: (e) lower exp, (E) upper exp, (f) fixed, (i) integer

Script Output: fprintf in Command Window

Example:

```
% This script prints out the surface area and volume  
% of a cylinder with a given radius and height
```

```
radius = 5;
```

```
height = 2;
```

```
surfarea = 2*pi*radius^2+ 2*pi*radius*height;
```

```
vol = pi*radius^2*height;
```

```
fprintf('For a cylinder with radius %i and height %e, the surface area  
is %5.2f and the volume is %5.3f', radius, height, surfarea, vol)
```

Script Output: fprintf to file

Three Steps

- 1 Opening a file with `fopen`

```
fid = fopen('filename','permission')
```

- permission: (r) reading, (w) writing, (a) appending

- 2 Writing output to open file with `fprintf`

```
fprintf(fid,'text')
```

- 3 Closing the file with `fclose`

```
fclose(fid)
```

Tips

- Saved in current directory
- Can write to several files using different `fid`, e.g. `fid1`, `fid2`
- Is vectorized \Rightarrow the command repeats itself until all elements are displayed (column-wise)

Script Output: fprintf to file

Example:

```
% This script creates a chart of inches to centimeters
```

```
inches = [1:10];
```

```
centim = 2.54*inches;
```

```
% Creates table to take care of vectorization
```

```
TBL = [inches;centim];
```

```
fid = fopen('in2cm.txt','w');
```

```
fprintf(fid, 'Inches to Centimeters Table\n\n');
```

```
fprintf(fid, 'Inches \t Centimeters\n');
```

```
fprintf(fid, '%i \t \t %3.2f\n', TBL);
```

```
fclose(fid);
```