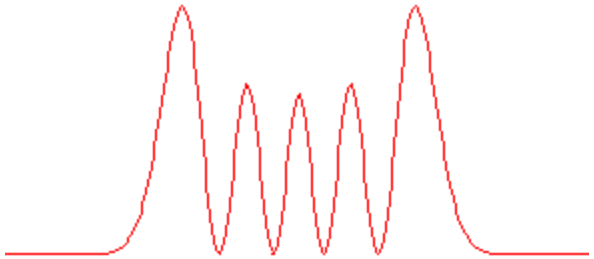
EP547 Computational Methods in QM

Topic 2

Array Manipulation in MATLAB



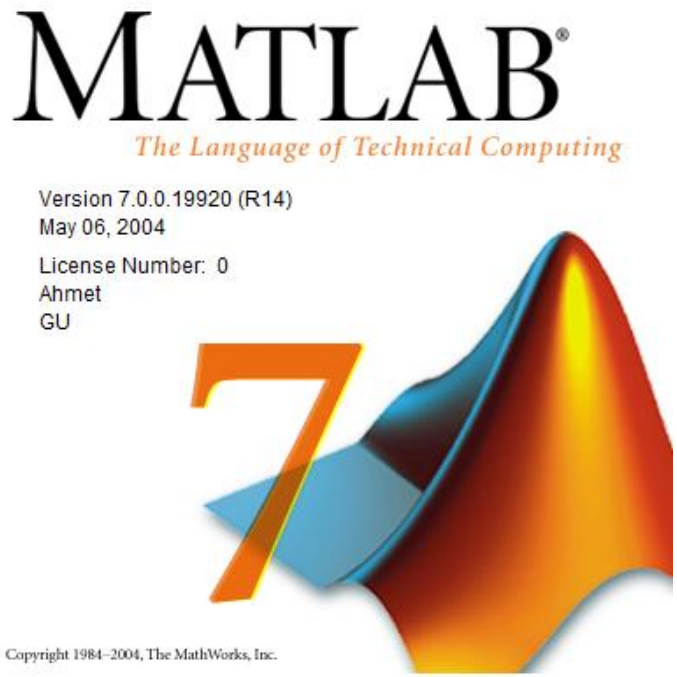
Department of Engineering Physics

University of Gaziantep

Feb 2013

Content

1. 1D Arrays
2. 2D Arrays
3. Array Functions
4. Reading Text files
5. Data Analysis



OneDim Arrays (Vectors)

- We learned before that an array can be created in may ways

```
>> x = [0  0.25  0.5  0.75  1]
x = 0    0.2500    0.5000    0.7500    1.0000
```

```
>> x = 0:0.25:1
x = 0    0.2500    0.5000    0.7500    1.0000
```

```
>> for i=1:5
     x(i) = (i-1)*0.25;
   end
>> x
x = 0    0.2500    0.5000    0.7500    1.0000
```

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

```
>> dizi2 = -5:2:5
dizi2 = -5    -3     -1     1     3     5
```

```
>> v = [1 2 3]    % row vector
v = 1   2   3
```

```
>> v = [1; 2; 3]   % column vector
v =
     1
     2
     3
```

```
>> v = [1 2 3]'   % transpose of a row vector
v =
     1
     2
     3
```

TwoDim Arrays (Matrices)

```
>> A = [1 1 1; 2 2 2]

A =  1     1     1
     2     2     2
```

```
>> A = [1 1 1
        2 2 2]

A =  1     1     1
     2     2     2
```

```
>> B = A'

B =  1     2
     1     2
     1     2
```

```
>> M = [9 8 7;  6 5 4;  3 2 1]
M =
     9     8     7
     6     5     4
     3     2     1
>> M(2,3)
ans = 4

>> M(:,3)
ans =
      7
      4
      1

>> M(2,:)
ans = 6     5     4

>> M(2,3) = 100
ans =
     9     8     7
     6     5   100
     3     2     1
```

Some Array Functions

length(x)	Number of elements of x
sum(x)	Sum the elements of x
mean(x)	Aritmetic mean of x
std(x)	Standard deviation of x
prod(x)	Product of the elements of x
dot(x,y)	Scalar product of x and y
cross(x,y)	Vector product of x and y
linspace()	
logspace()	
size(A)	number of rows and of matrix A
zeros(n)	
ones(n)	
eye(n)	
rand(n)	
magic(n)	

- **`linspace()`** function

```
>> x = 0:0.25:1
x = 0    0.2500    0.5000    0.7500    1.0000
```

```
>> x = linspace(0,1,5)
x = 0    0.2500    0.5000    0.7500    1.0000
```

- **`logspace()`** function: *logarithmic counterpart of linspace*

```
>> x = logspace(0,1,5)
x =  1.0000    1.7783    3.1623    5.6234   10.0000
```

creates 5 logarithmically spaced elements

starting with $x = 10^0$ and ending with $x = 10^1$.

```
>> x = [0  0.25  0.5  0.75 1];
>> length(x)
ans = 5
```

```
>> A = [1 1 1; 2 2 2];
>> size(A)
ans =  2   3
```

```
>> x = [1  2  2.5  3   3.1];
>> sum(x)
ans = 11.6000

>> prod(x)
46.5000
```

```
>> a = [1 2 4];
>> b = [0 2 5];
>> dot(a,b)
ans = 24
>> cross(a,b)
ans =  2    -5     2
```

zeros(m,n)	*returns a matrix of m rows and n columns that is filled with zeroes*
ones(m,n)	*returns a matrix of m rows and n columns that is filled with ones*
rand(m,n)	*returns a matrix of m rows and n columns that is filled with uniform random number between [0,1]*
eye(n)	*creates an n x n identity (unit) matrix.*

```
>> P = zeros(2,3)
P =  0     0     0
     0     0     0
>> P = ones(2,3)
P =  1     1     1
     1     1     1
>> P = rand(2,3)
P = 0.9501    0.6068    0.8913
    0.2311    0.4860    0.7621
>> I = eye(2)
I =  1     0
     0     1
```

Vectors and Relational Operators

- Recall the relational operators ==, ~=, <, >, <=, and >=.
- These operators can be applied to vectors element-wise to generate a same-sized vector of logical results.

```
>> X = [1 9 8 4];
>> Y = [4 3 0 4];
>> X>Y
ans = 0 1 1 0

>> X == Y
ans = 0 0 0 1

>> X>Y | Y == 4
ans = 1 1 1 1
```

```
>> X = [1 9 8 4];
>> Y = [4 3 0 4];
>> sum(X>Y)
ans = 2

>> sum(X == Y)
ans = 1

>> sum(X>Y | Y == 4)
ans = 4
```

Reading Text Files

Consider **gravity.txt** (*) contains 1000 measurements of gravitational acceleartion (g) on the Earth surface at sea level.

Find the mean, maximum and minimum values of the data.

We can read this data directly into a vector and process it as follows:

```
>> g = load ('gravity.txt');
>> mean(g)
ans = 9.8120
>> max(g)
ans = 10.4856
>> min(g)
ans = 9.0473
```

```
>> g = textread('gravity.txt');
>> mean(g)
ans = 9.8120
>> max(g)
ans = 10.4856
>> min(g)
ans = 9.0473
```

(*) Download the file at:

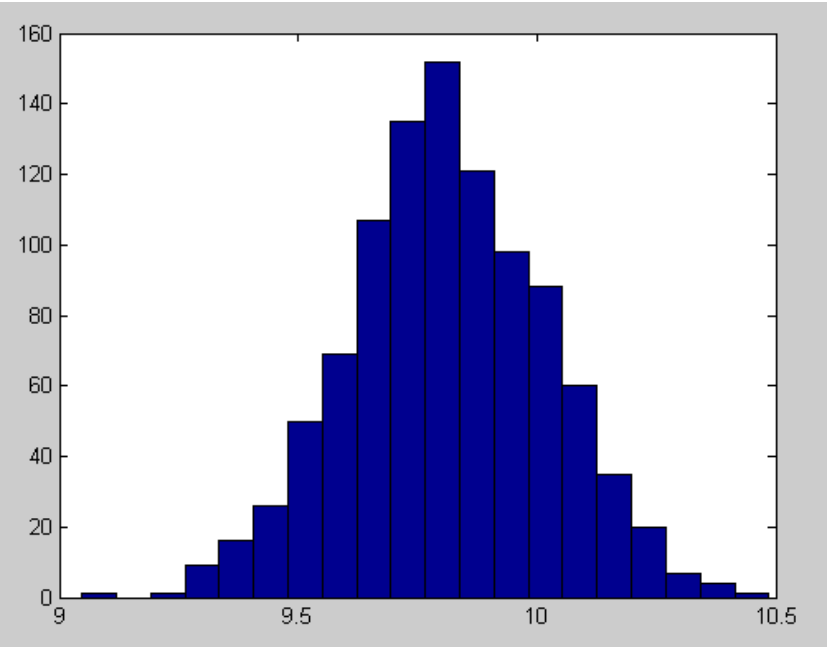
http://www1.gantep.edu.tr/~bingul/ep547/src/gravity.txt

Data Analysis

Data analysis is a very broad subject covering many techniques and types of data. In this lecture we will study some basic calculations that are commonly performed on sampled data.

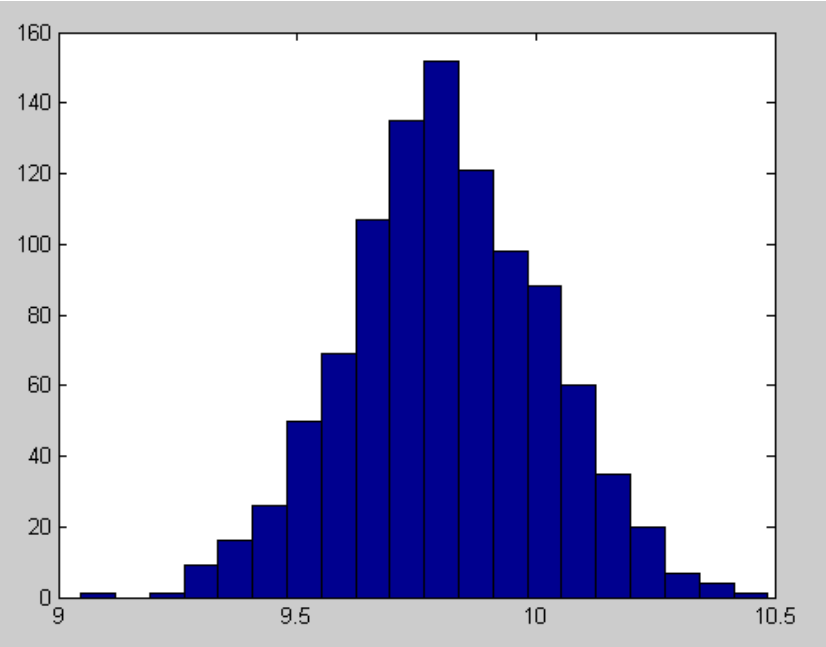
Consider again the file **gravity.txt**. We can plot the histogram of the data:

```
>> g = textread('gravity.txt');
>> hist(g,20)
```



The ***mean*** $\bar{x}$ of the sample is given by:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$



The ***standard deviation*** σ of the sample is defined as:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

```
>> g = textread('gravity.txt');
>> mean(g)
ans = 9.8120
>> std(g)
ans = 0.2077
```

For this data, σ is the size of the variation of *g* about the mean.

For bi-variate data (two variables) the correlation coefficient (ρ) is a measure of the linear dependence between one variable and the other.

Given a sample (size n) of bi-variate data,
$Z = \{ (x_1,y_1),\ (x_2,y_2),\ (x_3,y_3),\ \ldots ,\ (x_n,y_n)\ \}$

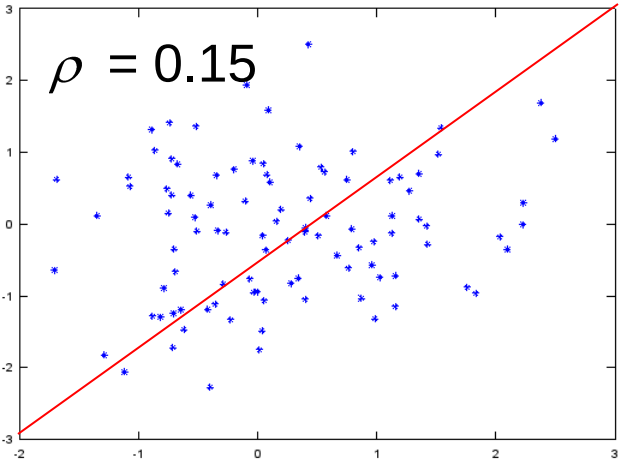
$$\rho = \frac{\overline{xy} - \bar{x}.\bar{y}}{\sigma_x \sigma_y}$$

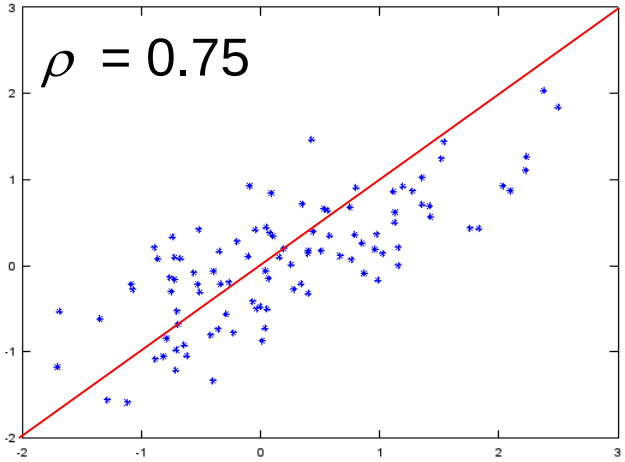
$$\overline{xy} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i \qquad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

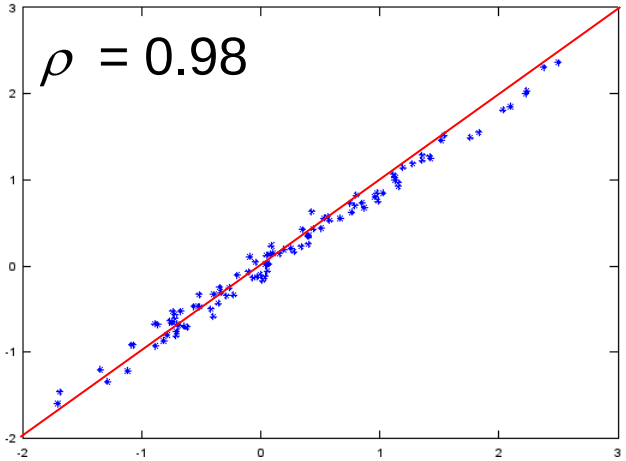
$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad \sigma_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\rho = \frac{\overline{xy} - \bar{x}.\bar{y}}{\sigma_x \sigma_y}$$

$$-1 \leq \rho \leq 1$$

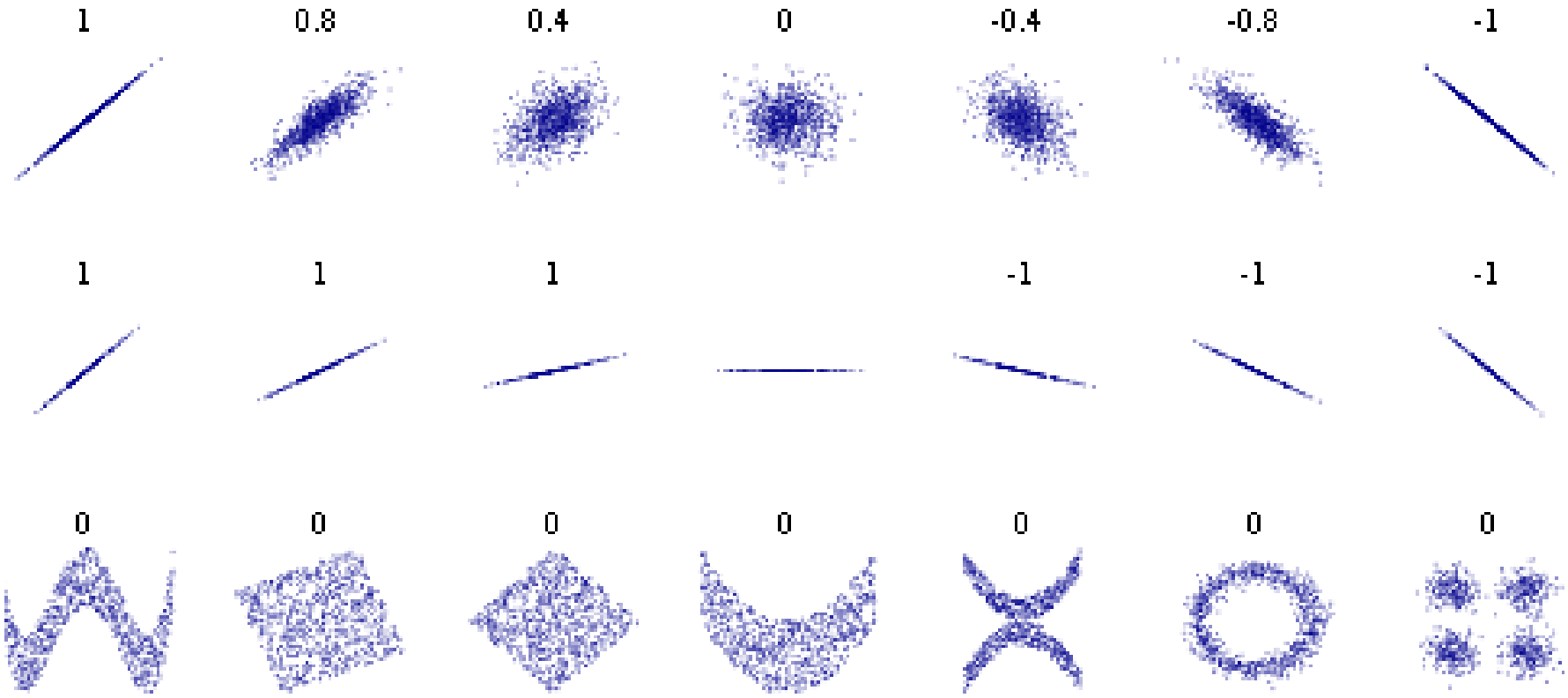






$\rho = 0$ if there is no correlation

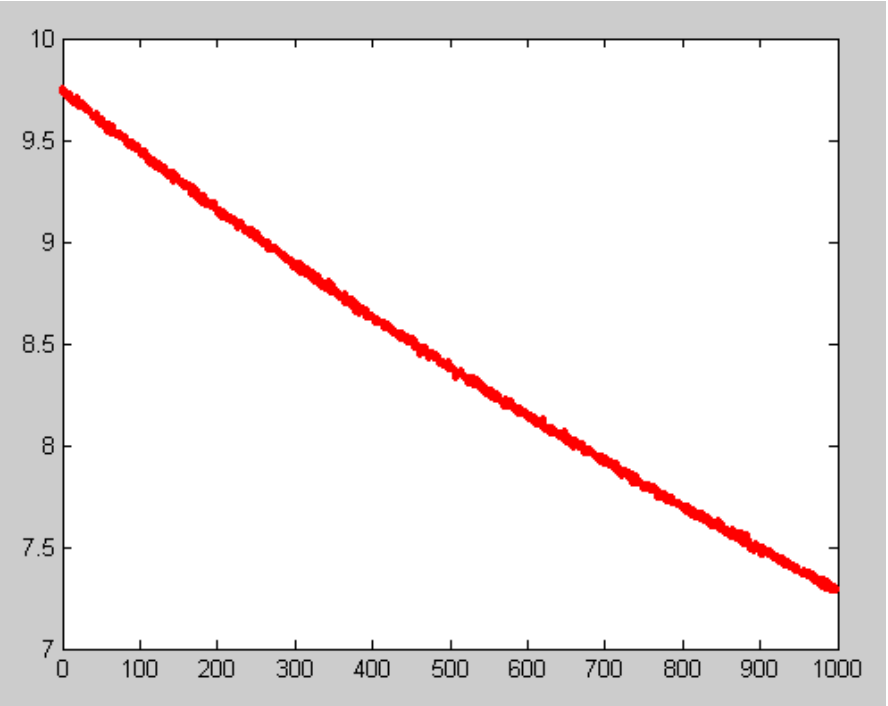
$\rho = \pm 1$ if X and Y are fully correlated


1
0.8
0.4
0
-0.4
-0.8
-1
1
1
1
-1
-1
-1
0
0
0
0
0
0
0

Now consider another file **gravity2.txt** (*) contains 1000 measurements of gravitational acceleartion (g) as a function of altitude (h) from Earth surface. First cloumn is the altitude in km and second column is measured value of g.

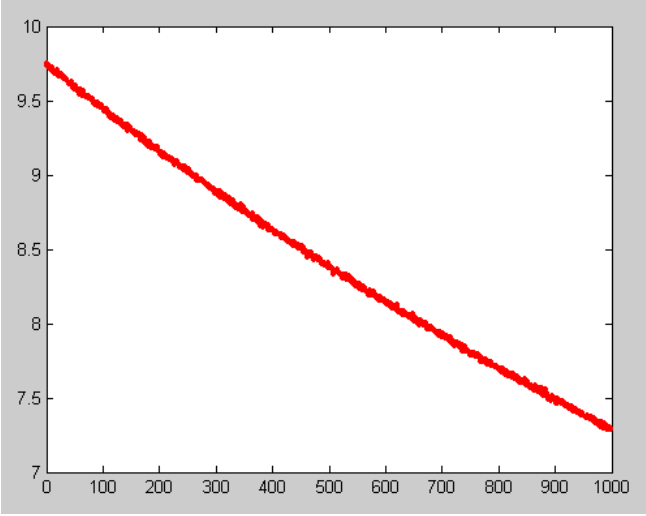
We can read this data directly into a vector and process it as follows:

```
>> data = load('gravity2.txt');
>> h = data(:,1);
>> g = data(:,2);
>> plot(h,g,'r.')
```



(*) Download the file at:
http://www1.gantep.edu.tr/~bingul/ep547/src/gravity2.txt

Correlation between h and g:




```
>> [x y] = textread('gravity2.txt');
>> plot(x,y,'r.')
>>
>> rho=(mean(x.*y)-mean(x)*mean(y))/(std(x)*std(y))
rho = -0.9974
```

There is clearly strong negative dependence on the altitude since $\rho \approx -1$. “A higher altidude leads to a lower gravity”.

References

[1]. http://www.mathworks.com/products/matlab

[2]. Numerical Methods in Engineering with MATLAB,
J. Kiusalaas, Cambridge University Press (2005)

[3]. Numerical Methods for Engineers, 6th Ed.
S.C. Chapra, Mc Graw Hill (2010)

[4]. http://wikipedia.org/wiki/Mean

[5]. http://wikipedia.org/wiki/Standard_deviation

[6]. http://en.wikipedia.org/wiki/Correlation