# EP375 Computational Physics

## Topic 10

## CURVE FITTING

**Department of Engineering Physics**

**University of Gaziantep**

**Feb 2013**

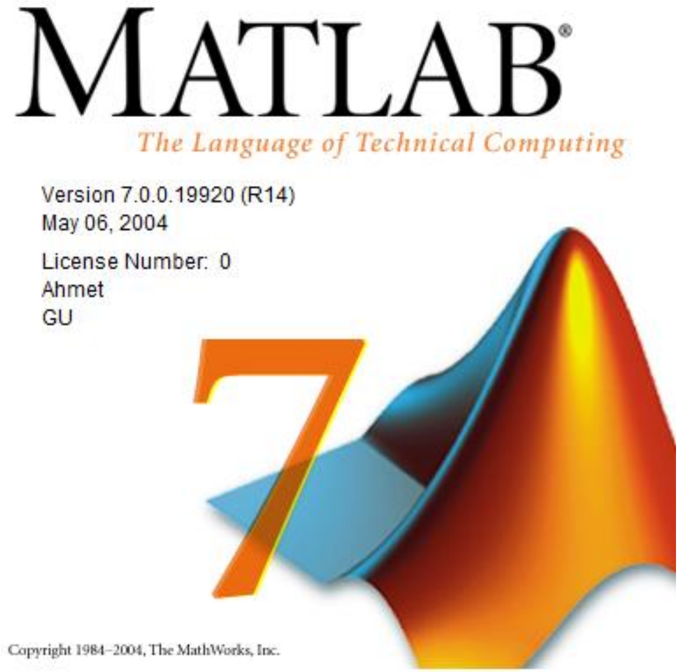
# Content

1. **Introduction**

2. **Least Square Method for Curve Fitting**

3. **MATLAB Functions and Tool**

4. **Example Applications**



# Introduction

- Data is often given for discrete values along a continuum.
- You may require estimates at points between discrete values.

*The distance required to stop an automobile is a function of its speed. The following data is collected to get this relationship:*

| v(km/h) | d(m) |
| --- | --- |
| 24 | 4.8 |
| 32 | 6.0 |
| 40 | 10.2 |
| 48 | 12.0 |
| 64 | 18.0 |
| 80 | 27.0 |

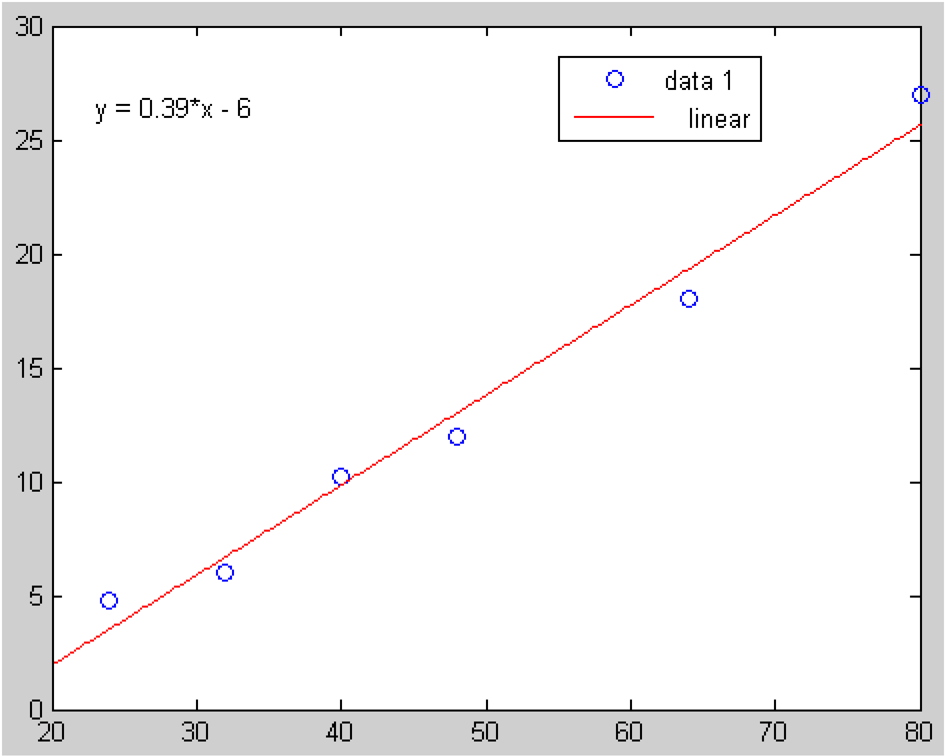
- In this section we will consider how to obtain values between the given experimental points using **Least square fitting method.**

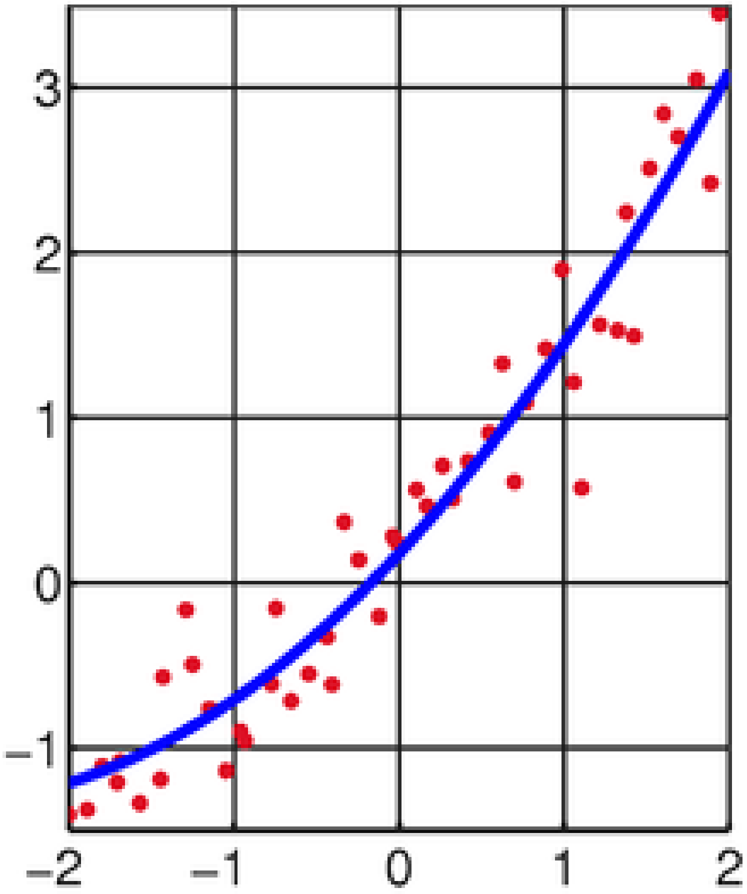
# Least Square Fitting Method

- The method of least squares is a standard approach to the approximate solution of over-determined systems, i.e. sets of equations in which there are more equations than unknowns.

- "Least squares" means that the overall solution minimizes the sum of the squares of the errors made in solving every single equation.

- Least squares problems fall into two categories: **linear** or **non-linear** least squares.



- The most important application is in data fitting.
- The best fit in the least-squares sense minimizes the sum of squared residuals defined by:

**residual = data – fit_model_function**

$$r_i = y_i - \hat{y}_i$$

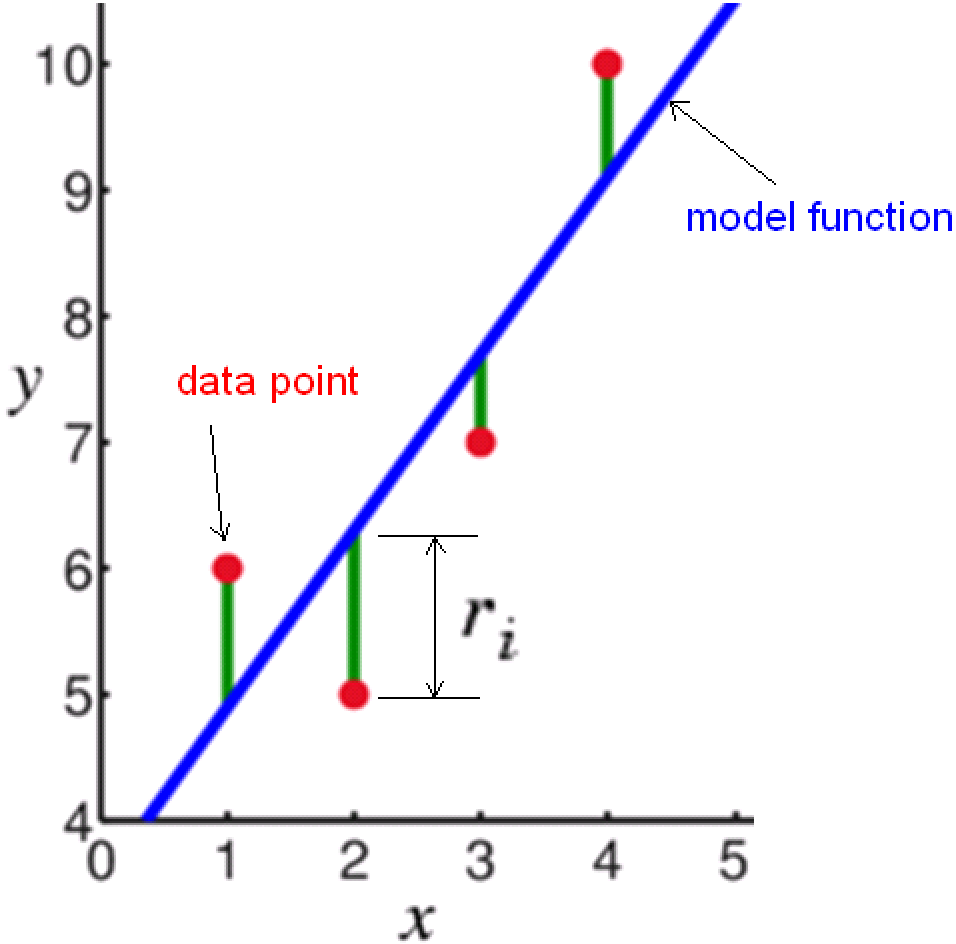
The summed square of residuals is given by

$$S = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

*Experimental data*

| x | y |
|---|---|
| $x_1$ | $y_1$ |
| $x_2$ | $y_2$ |
| . | . |
| . | . |
| . | . |
| $x_n$ | $y_n$ |

where ***n*** is the number of data points included in the fit and ***S*** is the sum of squares error estimate.




*Experimental data*

| x | y |
|---|---|
| $x_1$ | $y_1$ |
| $x_2$ | $y_2$ |
| . | . |
| . | . |
| . | . |
| $x_n$ | $y_n$ |

$$r_i = y_i - \hat{y}_i$$

# Linear Least Square Method (two parameters)

- Consider we want to fit the a data ($x_i$, $y_i$) to a function **y = ax + b** then the square sum of the residuals is:

$$S = \sum_{i=1}^{n} (y_i - a x_i - b)^2$$

- To minimize *S*, we should solve the following equations simultaneously:

$$\frac{\partial S}{\partial a} = 0 \qquad \frac{\partial S}{\partial b} = 0$$

- Solutions are:

$$a = \frac{n\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{n\sum x_i^2 - \left(\sum x_i\right)^2} \qquad b = \frac{\left(\sum x_i^2\right)\left(\sum y_i\right) - \left(\sum x_i\right)\left(\sum x_i y_i\right)}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

# Goodness of the Fit:

$$r^2 = \frac{S_t - S}{S_t}$$

where

$$S = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

$$S_t = \sum_{i=1}^{n} (y_i - y_m)^2$$

$$y_m = \frac{\sum_{i=1}^{n} y_i}{n}$$ (mean value of y)

For a good fit

$$S \to 0$$

$$r^2 \to 1$$

**Example 3**: *Linear Fit*

The distance required to stop an automobile is a function of its speed. The following data is collected to get this relationship:

| v(km/h) | d(m) |
|---|---|
| 24 | 4.8 |
| 32 | 6.0 |
| 40 | 10.2 |
| 48 | 12.0 |
| 64 | 18.0 |
| 80 | 27.0 |

Fit the data to a linear function and compute the goodness of the fit.

fitting.m

```
% get the data from the file
load data.txt
x = data(:,1);
y = data(:,2);
n = length(x);

% compute the coefficients
factor = n*sum(x.*x)-sum(x)*sum(x);
a = (n*sum(x.*y)-sum(x)*sum(y))/factor;
b = (sum(x.*x)*sum(y)-sum(x)*sum(x.*y))/factor;

% compute the goodness of the fit
ym = sum(y)/n;
S = sum((y-a*x-b).^2);
St = sum((y-ym).^2);
r2 = (St-S)/St;
% print out the results
fprintf('Fit Results\n');
fprintf('a = %f\n',a);
fprintf('b = %f\n',b);
fprintf('r2= %f\n',r2);
```

data.txt

```
24    4.8
32    6.0
40   10.2
48   12.0
64   18.0
80   27.0
```

*Command window*:

```
>> fitting
Fit Results
a = 0.394853
b = -5.952941
r2= 0.980287
```

# Weighted Linear Least Square Method

- Weighted least squares regression minimizes the error estimate:

$$S = \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2$$

where $w_i$ are the weights which determine how much each response value influences the final parameter estimates.

- If you know the variances of your data, then the weights are given by:

$$w_i = 1/\sigma_i^2$$

**Example 4**: *Weighted Linear Fit*

The distance required to stop an automobile is a function of its speed. The following data is collected to get this relationship:

| v(km/h) | d(m) |
|---|---|
| 24 | 4.8 +- 0.3 |
| 32 | 6.0 +- 0.4 |
| 40 | 10.2 +- 1.0 |
| 48 | 12.0 +- 1.1 |
| 64 | 18.0 +- 1.4 |
| 80 | 27.0 +- 1.5 |

The **+- value** represents the measurement error (one standard deviation).

Fit the data to a linear function and compute the goodness of the fit.

fitting.m

```
% get the data from the file
load data.txt
x = data(:,1);
y = data(:,2);
w = 1/data(:,3).^2;
n = length(x);
% compute the coefficients
...
```

data.txt

```
24    4.8  0.3
32    6.0  0.4
40   10.2  1.0
48   12.0  1.1
64   18.0  1.4
80   27.0  1.5
```

# Non-Linear Least Square Method (Generalized)

- In general, error estimate (*S*) can be written as:

$$S(\mathbf{a}) = \sum_{i=1}^{n} w_i (y_i - f(x_i, \mathbf{a}))^2 = \sum_{i=1}^{n} \left( \frac{y_i - f(x_i, \mathbf{a})}{\sigma_i} \right)^2$$

where **a** is a vector of coefficients: **a** = $\{a_1, a_2, \ldots, a_m\}$.

- To minimize *S*, we should solve the following *m* equations simultaneously:

$$\frac{\partial S}{\partial a_j} = 0 \qquad j = 1, 2, \cdots, m$$

- To get the solution, one can apply the following iterative method that we used before in Optimization:

$$\boxed{\mathbf{a}_{i+1} = \mathbf{a}_i - \mathbf{H}_i^{-1} \nabla S_i}$$

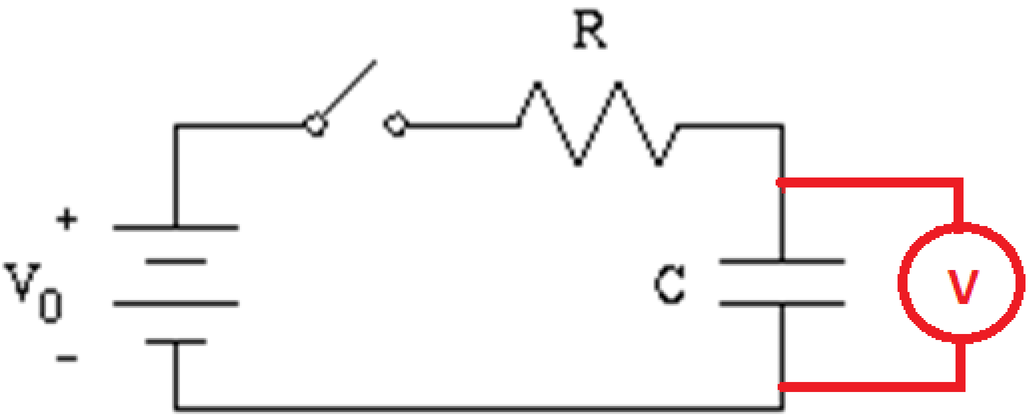
where

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \quad \mathbf{H} = \begin{pmatrix} \frac{\partial^2 S}{\partial a_1^2} & \frac{\partial^2 S}{\partial a_1 \partial a_2} & \cdots & \frac{\partial^2 S}{\partial a_1 \partial a_m} \\ \frac{\partial^2 S}{\partial a_2 \partial a_1} & \frac{\partial^2 S}{\partial a_2^2} & \cdots & \frac{\partial^2 S}{\partial a_2 \partial a_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 S}{\partial a_m \partial a_1} & \frac{\partial^2 S}{\partial a_m \partial a_2} & \cdots & \frac{\partial^2 S}{\partial a_m^2} \end{pmatrix} \quad \nabla S = \begin{pmatrix} \partial S / \partial a_1 \\ \partial S / \partial a_2 \\ \vdots \\ \partial S / \partial a_m \end{pmatrix}$$

## **Example***: *Non-Linear Fit*

Consider a charging RC circuit containing a resistance (R) and an initially uncharged capacitor (C). The switch is closed at $t$ = 0 and using a voltmeter the following experimental data is obtained. Each voltage measurement has constant 0.05 V error.

| t(sec) | Vc(Volts) |
|---|---|
| 0.5 | 2.7 |
| 1.0 | 4.7 |
| 1.5 | 6.3 |
| 2.0 | 7.6 |
| 2.5 | 8.6 |
| 3.0 | 9.3 |

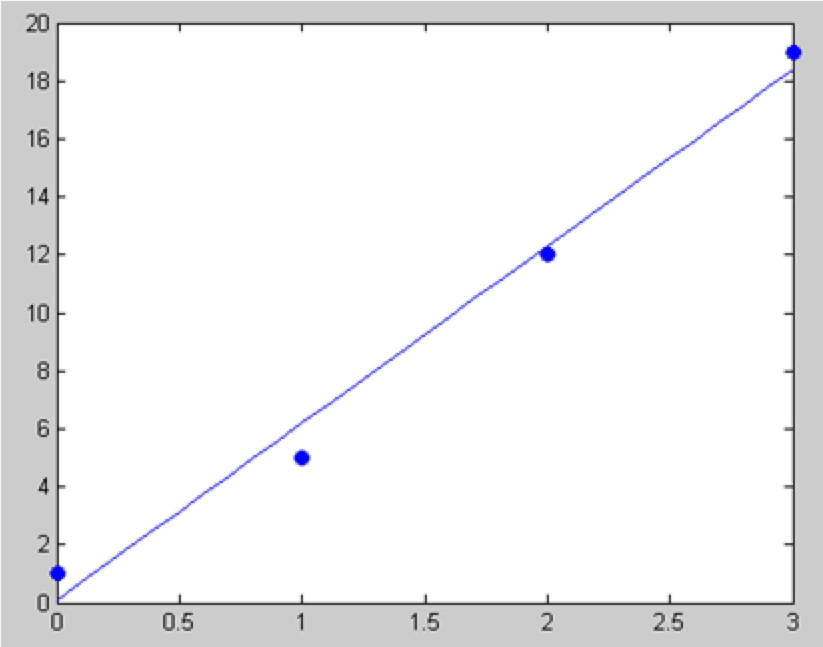



where $t$ is time and $Vc$ is potential difference the across the capacitor. Using least square fitting method, determine the time constant of the circuit if $V_0$ = 12 V.

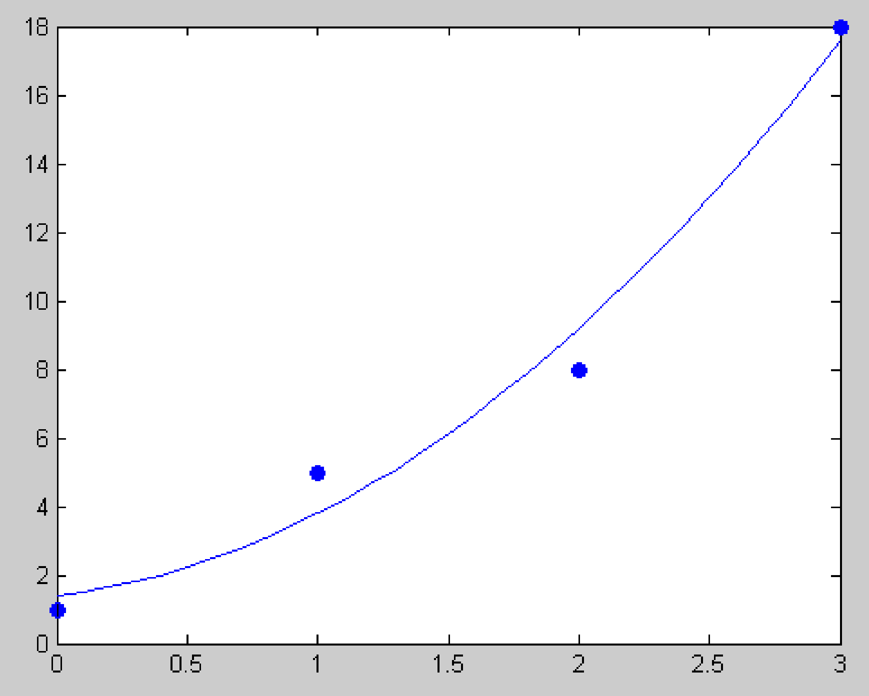
# Polynomial Fit

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

```
>> x = [0 1 2 3];
>> y = [1 5 12 19];
>> polyfit(x,y,1) % first order polynomial

ans =  6.1000    0.1000

>> xi = 0:0.1:3;
>> yi = 6.1*xi + 0.1;
>> plot(x,y,'o')
>> hold on
>> plot(xi,yi)
```



```
>> x = [0 1 2 3];
>> y = [1 5 8 18];
>> polyfit(x,y,2) % second order polynomial

ans = 1.5000    0.9000    1.4000

>> xi = 0:0.1:3;
>> yi = 1.5*xi.*xi + 0.9*xi + 1.4;
>> plot(x,y,'o')
>> hold on
>> plot(xi,yi)
```



# MATLAB Curve Fitting Tool

```
>> cftool
```

## HW 1:

Analytically, fit the given data into function

(a) $y = ax + b$

(b) $y = d\,x^2$

(c) $y = c_1 + c_2x + c_3x^2$

| x | y |
|---|---|
| -3 | 3 |
| 0 | 1 |
| 2 | 1 |
| 4 | 3 |

## HW 2:

Analytically, fit the given data into function

(a) $y = ax + b$

(b) $y = d\,x^2$

(c) $y = c_1 + c_2x + c_3x^2$

| x | y |
|---|---|
| -3 | 3 +- 0.3 |
| 0 | 1 +- 0.5 |
| 2 | 1 +- 0.5 |
| 4 | 3 +- 0.3 |

**HW 3**:

Write a matlab function of the form:

```
function tau = capacitor(t, Vc)



end
```

to return time constant of the circuit using least square fitting method for the same data in the Example*.

Here **t** and **Vc** are vectors of measurement.

## HW 4:

The table shows the experimental results of the measured Coulomb Force, $F$, between two charges, ($q_1$ and $q_2$) corresponding to distance $r$.

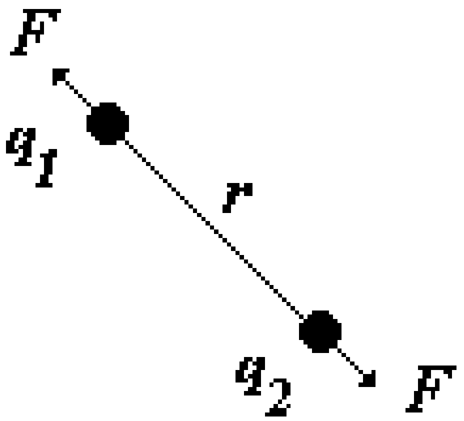
General form of the Coulomb Force is:

$$F(r) = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{r^n}$$

| r (cm) | F (N) |
|---|---|
| 40 | 2.5 +- 0.1 |
| 30 | 4.5 +- 0.1 |
| 20 | 10.1 +- 0.2 |
| 10 | 40.5 +- 0.4 |
| 5 | 162.0 +- 0.9 |

q1 = 5 µC
q2 = 9 µC

*Experimental setup*



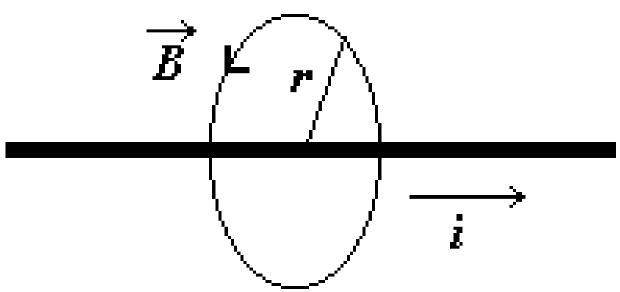

Determine value of $\varepsilon_0$ and $n$ using least-square method.

## HW 5:

*Magnetic field* around a long-wire carrying current *i* can be calculated from:

$$B(r) = \frac{\mu_0 i}{2\pi r}$$

where *r* is the distance measured form the wire and is the magnetic permeability constant. Table shows the experimental results of the measured magnetic fields, *B*, corresponding to the distance *r*.
The current is kept constant of 2.7 A and experimental set up is shown in Figure given below:




| r | B |
|---|---|
| 10.0 | 5.4 |
| 20.0 | 2.7 |
| 30.0 | 1.8 |
| 40.0 | 1.4 |
| 50.0 | 1.0 |

r is measured in cm
B is measured in μT
i is 2.7 A

Determine value of $\mu_0$ using least-square method.

## HW 6:

Table shows a data obtained from a radioactive substance. Here t is the time in seconds and *R* is the decay rate measured in Bq. Using least weighted square fitting method, determine the half life and the identity of the nucleus. Assume that each decay rate measurement (*R*) has an associated counting error of sqrt(*R*).

**e.g**, for *R* = 300 Bq then measurement error is sqrt(300) = 17.3 Bq.

| t(s) | R(Bq) |
|---|---|
| 40 | 300 |
| 100 | 245 |
| 140 | 210 |
| 200 | 165 |
| 240 | 127 |
| 300 | 110 |
| 340 | 90 |
| 400 | 85 |
| 440 | 58 |
| 500 | 45 |

# 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)