State University of New York at Buffalo

Mechanical and Aerospace Engineering Department

MAE 550: OPTIMIZATION IN ENGINEERING DESIGN

FINAL PROJECT

NAME: BABAK FARROKH
SEUNG KOOK JUN
LENG FENG LEE

DATE: 14 DEC 2002.

1. ABSTRACT.

In this project, we implemented numerical optimization techniques in minimizing the error in a bi-polynomial (A cubic and a quadratic polynomial) curve fitting method for the magnetostrictive response of thin film under the applied magnetic field. We first formulate the objective function using the least square method, and our objective is to minimize the norm of residual. The constraints were then formulated using the continuity requirement for the curve fitting and finally, we implemented MatLab Optimization ToolBox in solving our optimization problem. Our optimization problem turns out to be an 8-dimensional non-linear minimization problem. The uniqueness of this model is that we are able to visualize the result even the problem is an 8 dimensional optimization problem. The numerical result was then presented and graphs showing the curve fitting were also plotted.

2. INTRODUCTION.

In engineering practice, it is often involved the processing of data gained from experiments in order to obtain useful engineering judgment. Data explained the behavior of dependent variables changed with respect to the independent variables. Often, engineer wish to use equations to describe this changes so that we have some physical insight to the problems. As a result, curve fitting of data is important in engineering practice and as such, in this project, we implement numerical optimization techniques that we have learned in order to minimize the error of curve fitting and get the best result. Obviously, numerical optimization techniques have a wide range of application in this area. We thus focus our problem in minimize the error in the curve fitting using a bi-polynomial function in an experiment that study the magnetostrictive response of a thin film under the applied magnetic field.

Specifically, in the following session, we covered the following topic:

1. Different type of curve fitting methods.
2. Introduction to the experiment and data collected.
3. Experiment data curve fitting requirements.
4. Optimization problem formulation.
5. Solving the optimization problem.
6. Result obtained.

3. DIFFERENT TYPE OF CURVE FITTING METHODS.

3.1. Introduction.

Curve fitting is an important and useful technique for handling data because it produces a mathematical model of the data that can compactly contain and represent its primary properties. It is a method for mathematically describing the relationship between two variables.

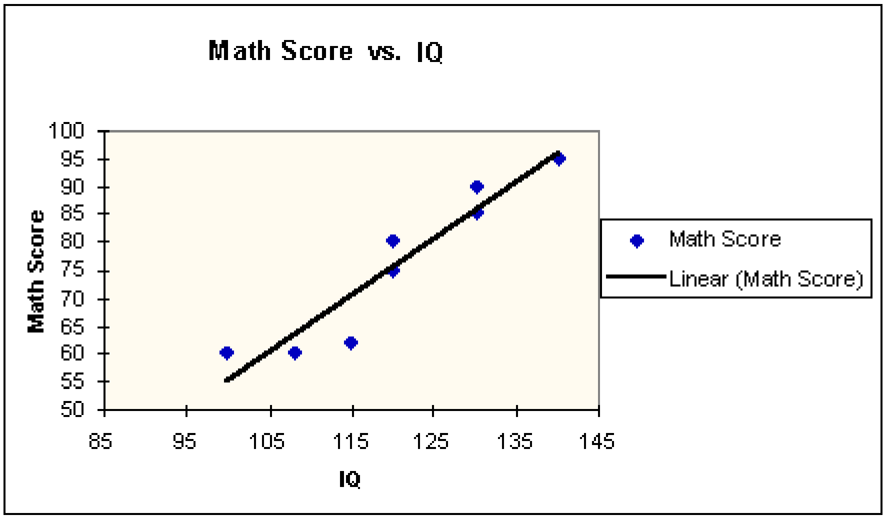
Curve fitting is an important and useful technique for handling data because it produces a mathematical model of the data that can compactly contain and represent its primary properties. Other words, by using curve fitting technique we are able to describe the relationship between two variables (for the problem in hand, Magnetic field or Electric field – associated induced strain) and then obtained equation provides a valuable tool for: predicting or forecasting future values of the dependent variable, interpolating measured values.

3.2. Possible relation of variables.

The relationship between data points determines the structure of the curve for each set of data. The relations are basically linear and nonlinear.

3.2.1 Linear Relation.

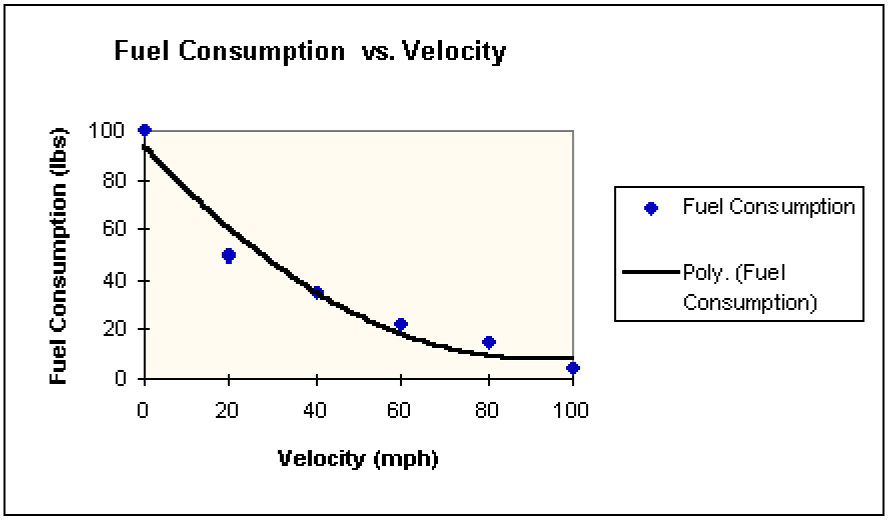
When a straight line best represents the data points, the mathematical expression showing their relationship is: $\boldsymbol{y = mx + b}$. For this type of relationship, it is said that the two variables are linearly related.



Examples of Linear Relationships: Math test scores vs. IQ (above) ; Current & voltage in an ideal resistor; Distance & time at constant velocity

3.2.2. Nonlinear relation.

Relationships between some data sets are nonlinear, e.g., quadratic or exponential. For this type of relationship, a nonlinear function best represents the data points.



Example of Nonlinear Relationship: Fuel consumption vs. Velocity (above)

3.3. METHODS FOR CURVE FITTING

There are three methods used to determine the parameters of a curve that best represents, fits, a set of data points.

3.3.1. Method of Selected Points

This is accomplished by visually selecting a curve that goes through as many data points as possible. This is the least accurate of the three methods. [4]

3.3.2. Method of Averages

This is based on the idea that a curve is positioned so that the algebraic sum of the differences between the observed and calculated values of the ordinates (the residuals) is equal to zero.

3.3.3. Method of Least Squares

This method is similar to the Method of Averages in the sense that it minimizes an error to get the best fit. In this case the error is the squared distance from each point to the curve. This third method is the most accurate.

Note: Since the Method of Least Squares is the most accurate method, and most of the curve/data fitting programs (e.g. Matlab) apply this method in order to fit a set of data, we briefly discuss this method (restricted to polynomial approximation) in next section.

3.4. Least Squares Data Fitting (Approximation)

In least squares technique, the approximated fit yields a polynomial that passes through the set of points in the *best possible manner* without being required to pass exactly through any of the points. Several definitions of "best possible manner" exist. Consider the set of discrete points $[x_i, Y(x_i)] = (x_i, Y_i)$ and the approximate polynomial $y(x)$ chosen to represent the set of discrete points, as illustrated in Figure 1.

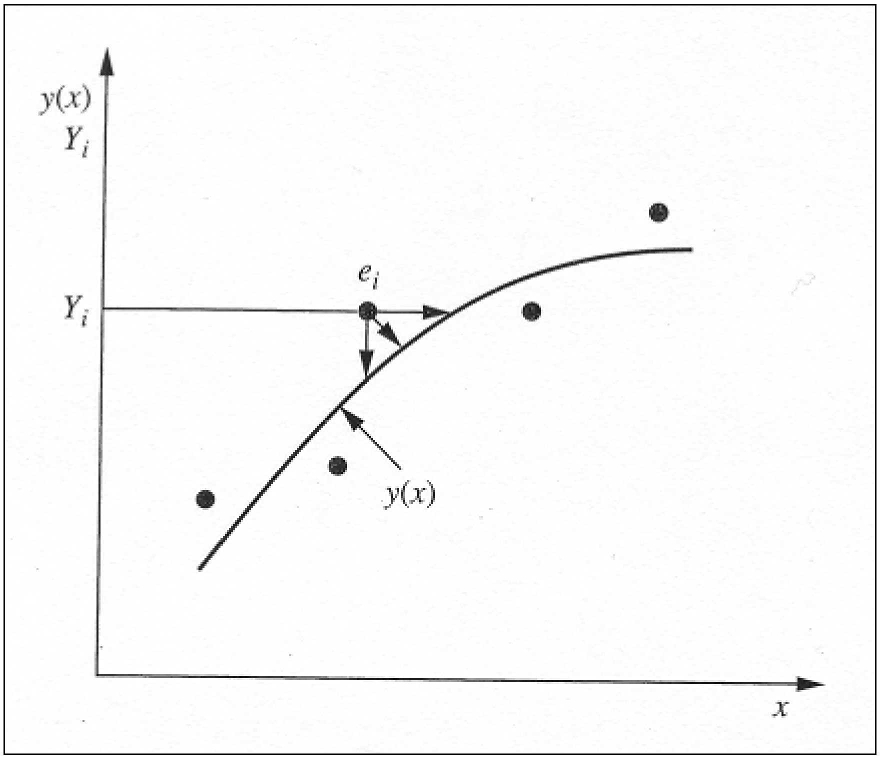


Figure 1. Approximate fit.

The discrete points do not fall on the approximating polynomial. The deviations (i.e. distances) of the points from the approximating function must be minimized in some manner. Some ambiguity is possible in the definition of the deviation. For example, if the values of the independent variable x_i are considered exact, then all the deviation is assigned to the dependent variable Y_i, and the deviation e, is the vertical distance between Y, and $Y_i = f(x_i)$. Thus,

$$e_i = Y_i - y_i \qquad \text{[Equation 1]}$$

It is certainly possible that the values of Y_i are quite accurate, but the corresponding values of x_i are in error. In that case, the deviation would be measured by the horizontal distance illustrated in Figure 1. If x_i and y_i both have uncertainties in their values, then the perpendicular distance between a point and the approximating function could be defined as the deviation. The usual approach in approximate fitting of tabular data is to assume that the deviation is the vertical distance between a point and the approximating function, as specified by Eq. (1).

Several best-fit criteria exist, as illustrated in Figure 2. for a straight-line approximation. Figure 2.a illustrates the situation where the sum of the deviations at two points is minimized. Any straight line that passes through the midpoint of the line segment connecting the two points yields the sum of the deviations equal to zero. Minimizing the sum of the absolute values of the deviations would yield the unique line that passes exactly through the two points.

That procedure also has deficiencies, however, as illustrated in Figure 2.b, where two points having the same value of the independent variable have different values of the dependent variable. The best straight line obviously passes midway between these two points, but any line passing between these two points yields the same value for the sum of the absolute values of the deviations. The minimax criterion is illustrated in Figure 2.c, where the maximum deviation is minimized. This procedure gives poor results when one point is far removed from the other points. Figure 2.d illustrates the least squares criteria, in which the sum of the squares of the deviations is minimized. The least squares procedure yields a good compromise criterion for the best-fit approximation.

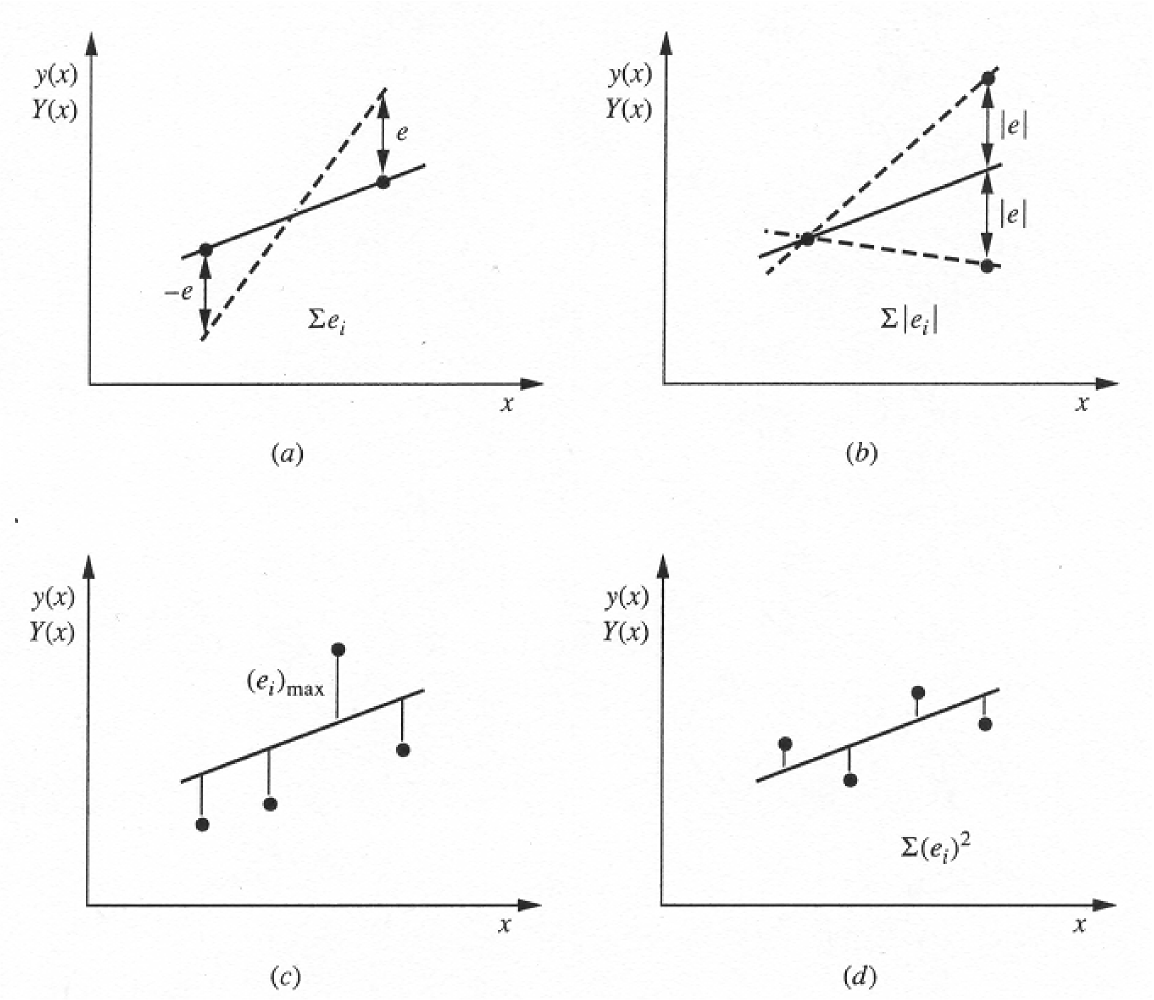


Figure 2. Best fit criteria: (a) minimize $\sum e_i$ *; (b) minimize* $\sum |e_i|$ *(c) minimax; (d) least square.*

The least squares procedure is defined as follows:

Given *N* data points $[x_i, Y(x_i)] = (x_i\ Y_i)$, choose the functional form of the approximating function to be fit, y = y(x), and minimize the sum of the squares of the deviations, $e_i = (Y_i - y_i)$. The least square method used in straight line approximation and higher order approximation were describe in the following session.

A.3. The Straight-Line Approximation

The simplest polynomial is a linear polynomial, the straight line. Least squares straight-line approximations are an extremely useful and common approximate fit. The least squares straight-line fit is determined as follows. Given N data points (x_i, Y_i), fit the best straight line through the set of data. The approximating function is:

$$y = b + mx \qquad \text{[Equation 2]}$$

At each value x_i, Eq. (2) gives

$$y_i = (b + mx_i) \qquad (i = 1,\ldots\ldots\ldots, N) \qquad \text{[Equation 3]}$$

The deviation e_i ar each value of x_i is

$$e_i = (Y_i - y_i) \qquad (i = 1,\ldots\ldots\ldots, N) \qquad \text{[Equation 4]}$$

The sum of squares of the deviation defines the function S(b, m):

$$S(b, m) = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} (Y_i - b - mxi)^2 \qquad \text{[Equation 5]}$$

The function S(b, m) is a minimum when $\frac{\partial S}{\partial b} = \frac{\partial S}{\partial m} = 0$. Thus,

$$\frac{\partial S}{\partial b} = \sum_{i=1}^{N} 2(Y_i - b - mx_i)(-1) = 0 \qquad \text{[Equation 6]}$$

$$\frac{\partial S}{\partial m} = \sum_{i=1}^{N} 2(Y_i - b - mx_i)(-1) = 0 \qquad \text{[Equation 7]}$$

dividing by 2 and rearranging yields,

$$bN + m\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} Y_i \qquad \text{[Equation 8]}$$

$$b\sum_{i=1}^{N} x_i + m\sum_{i=1}^{N} x_{\mathrm{i}}^2 = \sum_{i=1}^{N} x_i Y_i \qquad \text{[Equation 9]}$$

4.4. HIGHER-DEGREE POLYNOMIAL APPROXIMATION

The least square procedure developed in the preceding subsection can be applied to higher–degree polynomials. Given the N data points (x_i, Y_i), fit the best nth-degree polynomial through the set of data. Consider the nth-degree polynomial:

$$Y = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n \qquad \text{[Equation 10]}$$

The sum of deviations is given by

$$S(a_0, a_1, \ldots, a_n) = \sum_{i=1}^{N}(e_i)^2 = \sum_{i=1}^{N}(Y_i - a_0 - a_1x_1 - \cdots - a_nx^n)^2 \qquad \text{[Equation 11a]}$$

$$\vdots$$

$$a_0\sum_{i=1}^{N}x^n{}_i + a_1\sum_{i=1}^{N}x^{n+1}{}_i + \cdots + a_n\sum_{i=1}^{N}x^{2n}{}_i = \sum_{i=1}^{N}x^n{}_iY_i \qquad \text{[Equation 11a]}$$

Equations (11) can be solved for a_0 to a_n by Gauss elimination.

We define *The Norm of The Residual* as $R = \sqrt{S(a_0, a_1, \cdots, a_n)}$ and this indicates the overall degree of fit: *lower R implies better fit.*

A problem arises for high-degree polynomials. The coefficients N to $\sum x_i^{2n}$ in Eq.11 vary over a range of several orders of magnitude, which gives rise to an ill-conditioned system. Normalizing each equation helps the situation. Double-precision calculations are frequently required. Values of n up to 5 or 6 generally yield good results, values of n between 6 and 10 may or may not yield good results, and values of n greater than 10 generally yield poor results.

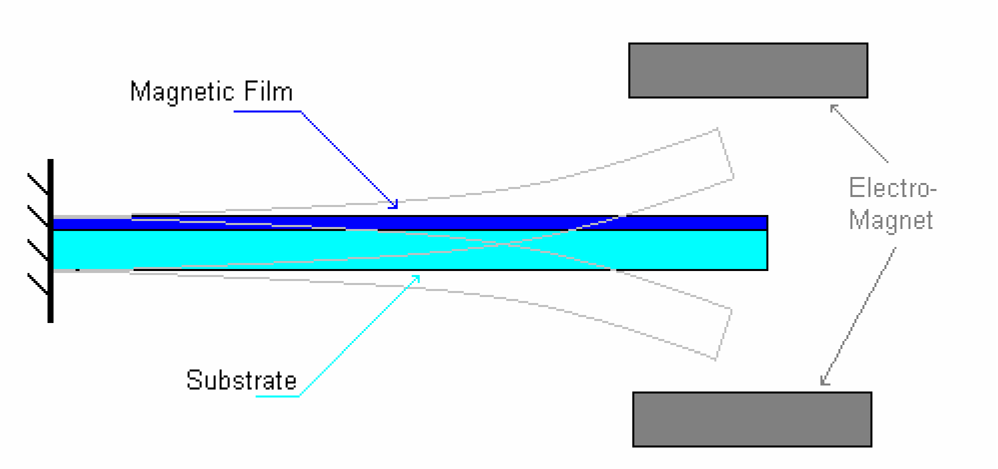
Preceding procedure for least square data fitting can be generalized and associated error (Norm of Residual) can be obtained as follows:

Assuming that there is a set of measured data as; $X_i = \{X_1, X_2, \cdots, X_n\}$ and $Y_i = \{Y_1, Y_2, \cdots, Y_n\}$. A function of the form $y_i = f(X_i)$, (e.g. $y_i = A \times a\tan(BX_i)$) can be fitted to these data in the least squares sense; similarly, in this case *The Norm of Residual* is defined as:

$$R = \left(\sum (y_i - Y_i)^2\right)^{\frac{1}{2}}$$

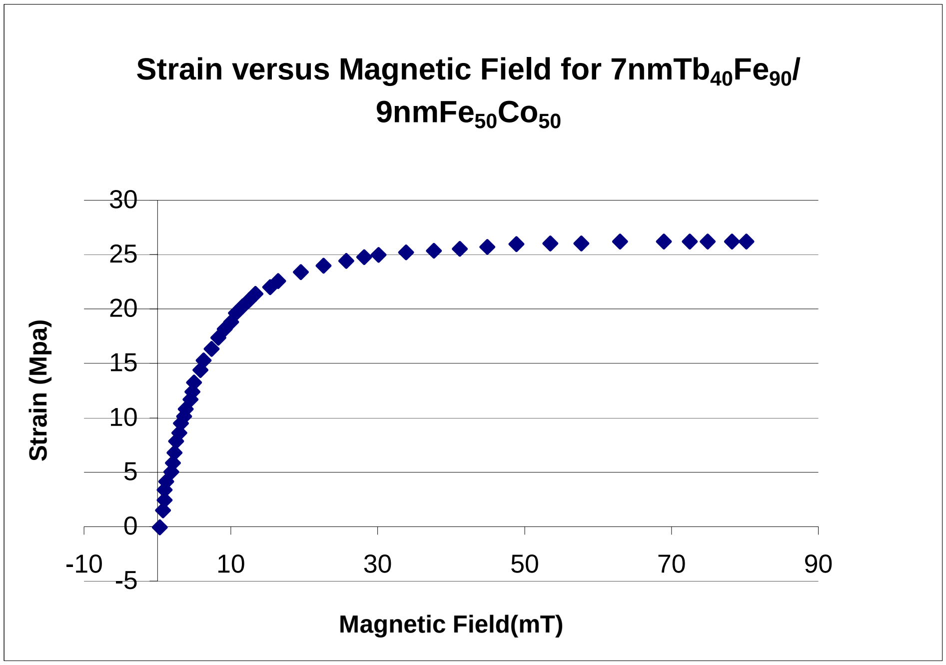
Optimization Problem formulation

Problem Description



- Magnetostrictive thin film under applied magnetic field -

This work is based on an experiment done by Alfred Ludwig and Eckhard Quant, where they measured the deflection response of giant magnetostrictive thin films of several magnetostrictive materials to the applied (positive/negative) magnetic field.

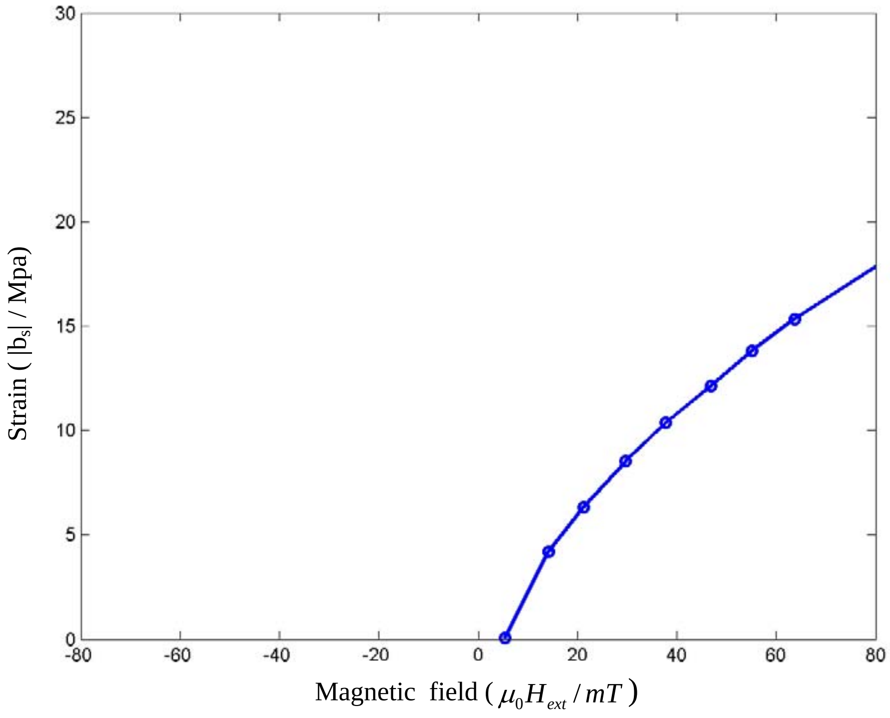


- Experiment Raw Data -

Generalization of the problem

There are three general types of behavior that these measured data represent, which can be listed as following:

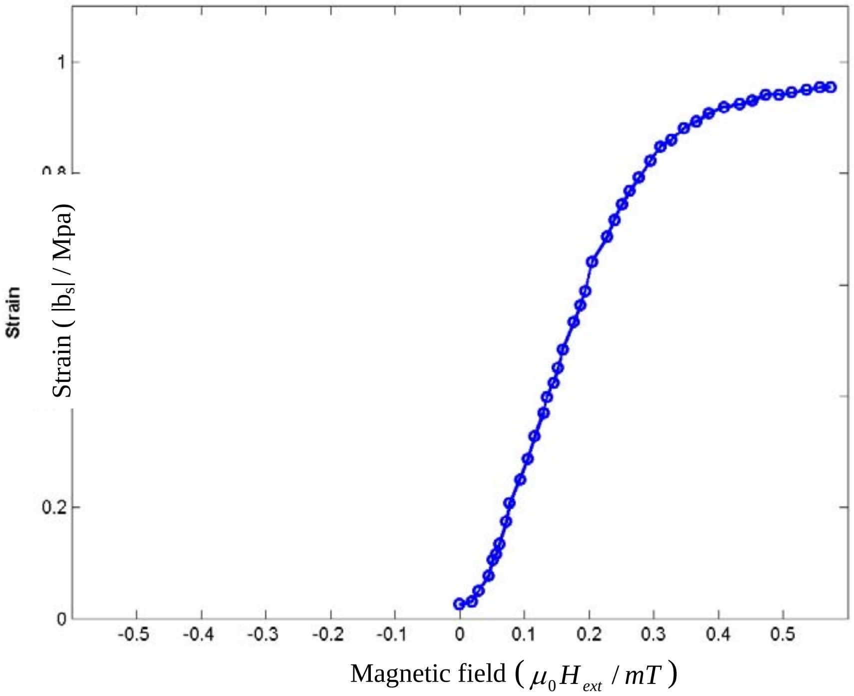
Type 1 : In some cases, measured strain response begins with a smooth slope and decreases smoothly as magnetic field increases. This relatively smooth response depends on the nature of these materials, where so-called, ‘*magnetization vector*’ of this materials does not rotate easily and resists against applied magnetic field, therefore a small change in slope (along the data points) deduced from this non-easiness in magnetization vector rotation. An example of this behavior indicated in figure below:



General response of Type 1.

For the cases similar to above, we can easily fit the data with a polynomial of low degree (up to 4th degree for the worst cases).

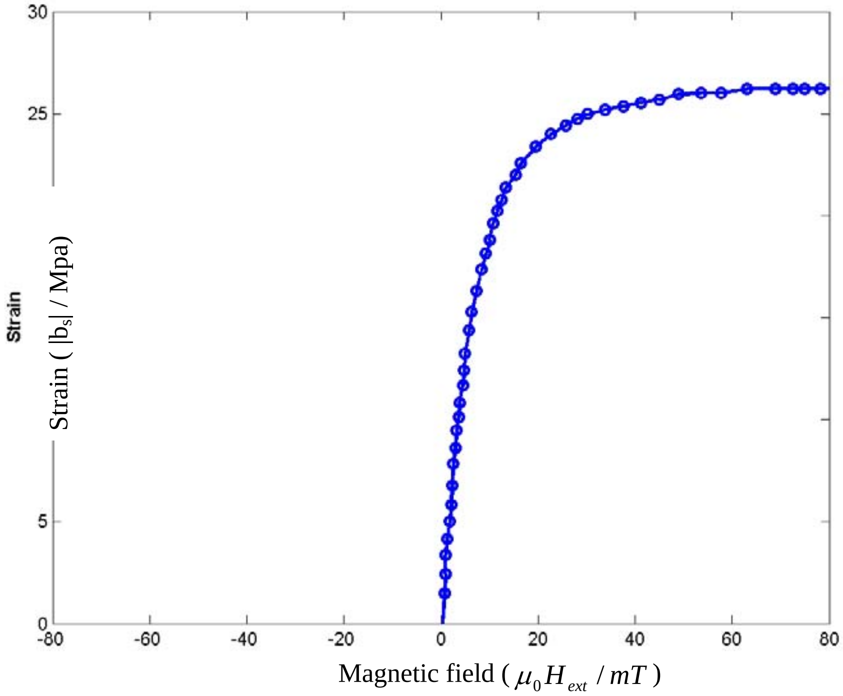
Type 2**:** In this case the initial slope is much greater than Type 1. This initial slope remains constant up to the point, which measured data inter into the saturated region, at this point a great change in slope toward zero can be observed. Comparing this kinds of behavior with type 1, we can state that the magnetization vector in this case tends to rotate much easier (compare two Type 1), namely by applying a small magnetic field we get a relatively large value of displacement, but as it gets closer to the saturation point it is harder to rotate the magnetization vector and slope starts decreasing very fast. A sample of this type of behavior indicated in figure below.



General response of Type 2.

The best possible model to be fitted to this type of response is a model of Cubic-Quadratic.

Type 3**:** In this case the initial slope is even greater than Type.2, this initial slope remains constant up to the point where measured data enters the saturated region. At this point a great change (greater than type 2) in slope toward zero can be observed. This case is similar to type 2 with the difference that for this type the magnetization vector rotates even easier that type 2 which deduces a steeper slope for the unsaturated region. A sample of this type behavior indicated in figure below:



General response of Type 3.

For this type of behavior Cubic-Quadratic gives the best result (less error).

Problem Formulation

It should be mentioned that Matlab Optimization Toolbox has been used in order to fit the different set of data with the above models.

As we already mentioned, the Cubic-Quadratic model gives the best result (Minimum error) in most of the cases, so this method will be discussed in more details as follows:

The problem is to fit a bi-polynomial model (Cubic-Quadratic) to a set of data. In particular, we would like to model the data with a combination of cubic-quadratic equation with the property of having continuity C_0 and the same slope (C_1 continuity) at the joint point (where cubic and quadratic polynomial intersect).

Quadratic equation: $Y_1 = A + Bx + Cx^2 \quad \rightarrow \quad \frac{dY_1}{dx} = B + 2Cx$

Cubic equation: $Y_2 = D = Ex + Fx^2 + Gx^3 \quad \rightarrow \quad \frac{dY_2}{dx} = E + 2Fx + 3Gx^2$

Constraint # 1: $A + Ba + Ca^2 = D + Ea + Fa^2 + Ga^3$: *C_0 continuity requirement* where ‘a’ is defined as x-coordinate of joint point.

Constraint # 2: $B + 2Ca = E + 2Fa + 3Ga^2$: *C_1 continuity requirement*

In order to satisfy these two constraints and still get error as less as possible, we introduce following 7-D non-linear Minimization Problem:

Minimize: $R \equiv$ *The Norm of The Residual.*

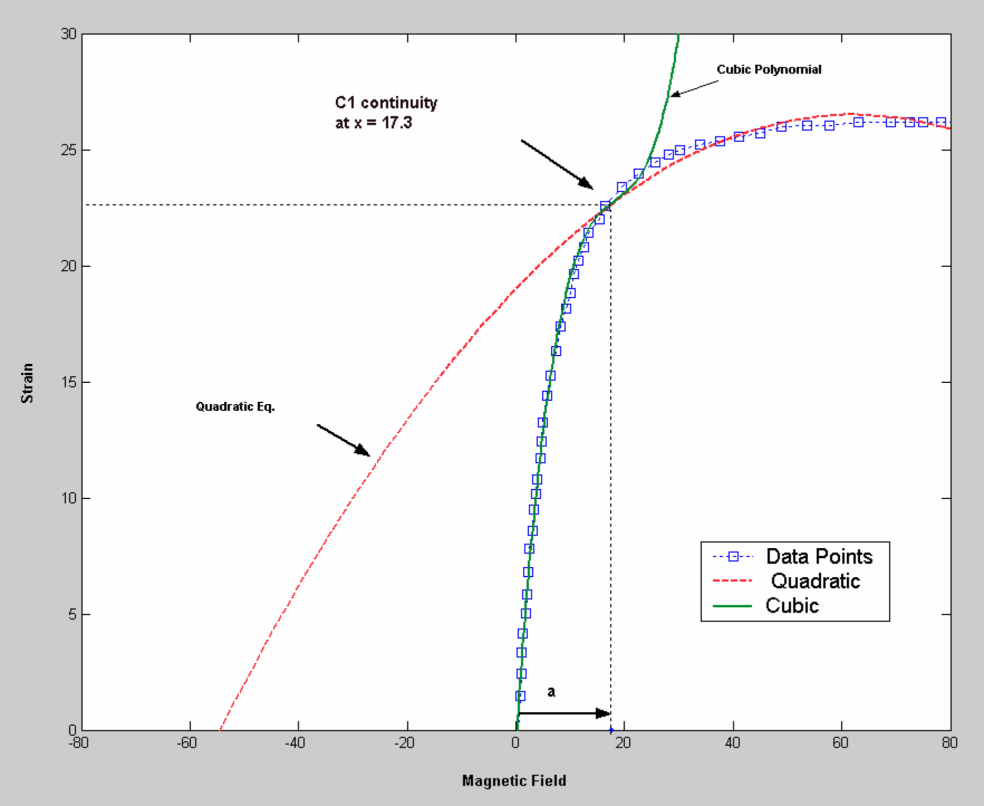
Subject to:

Constraint #1: $A + Ba + Ca^2 - D + Ea + Fa^2 + Ga^3 = 0$

Constraint # 2: $B + 2Ca - E + 2Fa + 3Ga^2 = 0$

Where; A, B, C, D, E, F and ‘a’ are design variables.

By solving the above optimization problem (using Matlab Optimization Toolbox) and estimating all seven design variables, which are the coefficients of cubic and quadratic equations, as well as the x-coordinate of the joint point, we are able to identify our model and get the value for the error associated with it.



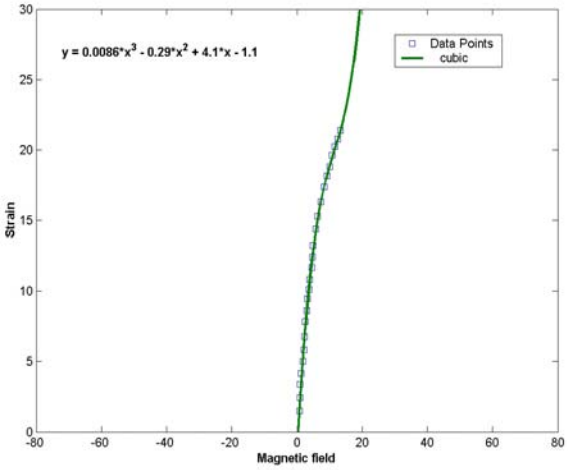
An illustration of Cubic-Quadratic model.

Where point A is the point where cubic and quadratic equations intersect with the same slope so that we minimize the Norm of Residual.

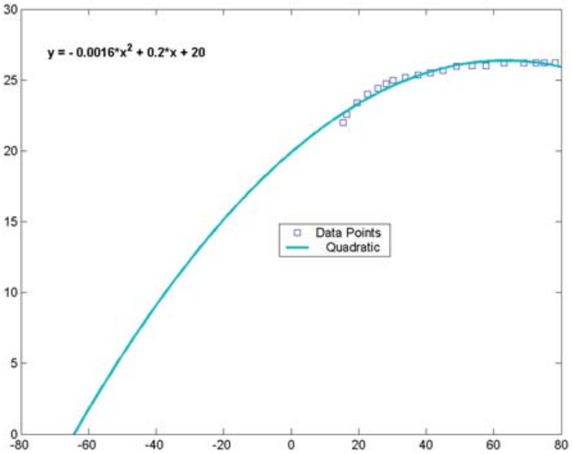
Note that the location of this point is one of our design variables as illustrated in above figure.

Problem Solving

1) Initialize Design Variables

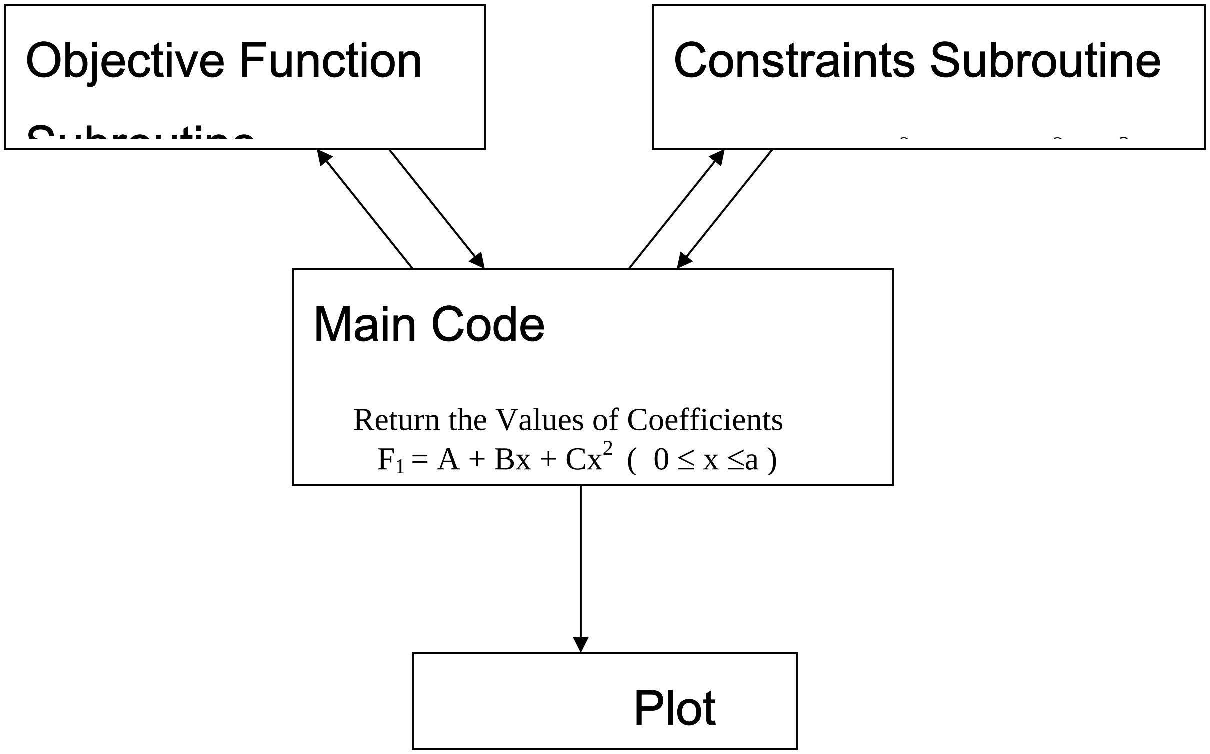


Approximately, we choose a portion of these data set that we guess could be fitted well with a cubic polynomial.
By doing these we will get a good initial guess for our DV’s; D,E,F and G.

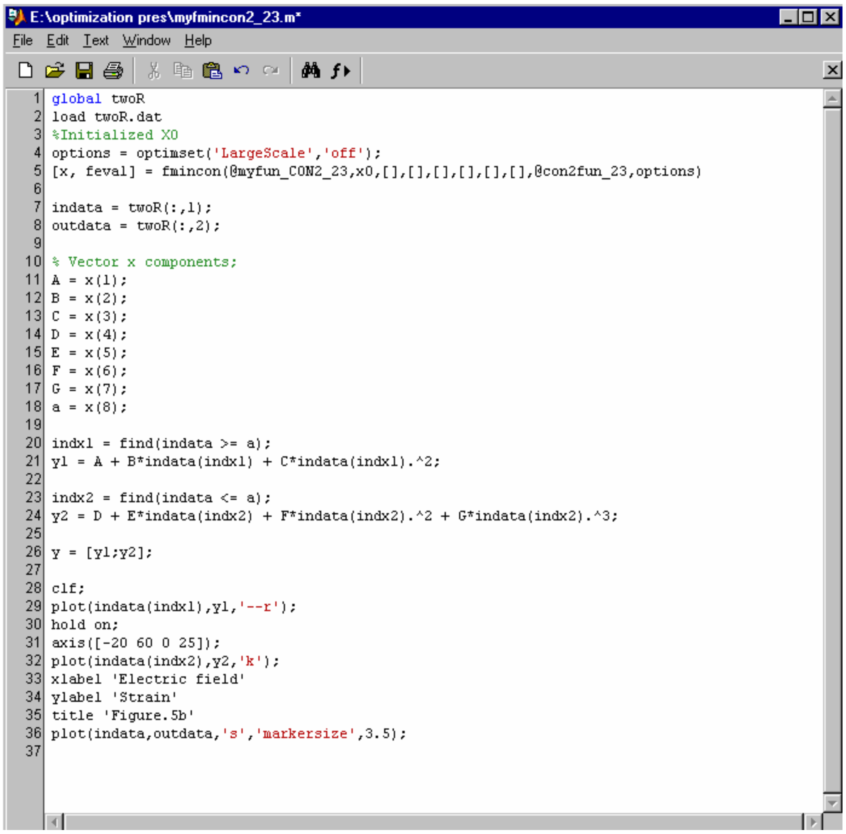


Similarly, following the same procedure for the upper portion of data points; the DV’s; A, B and C could be initialized

2) Code Diagram (Using Matlab Optimization tool box)

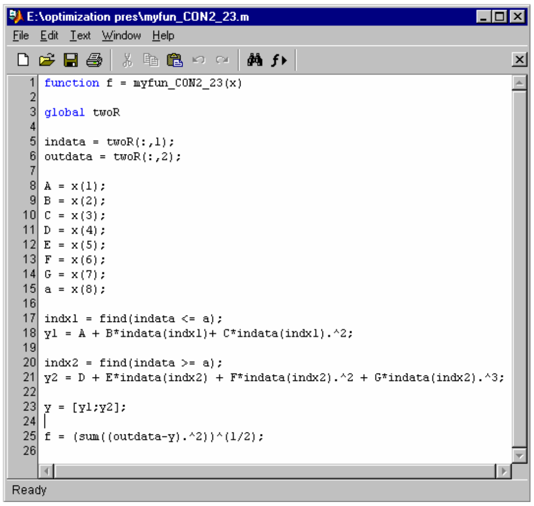


The Main Code



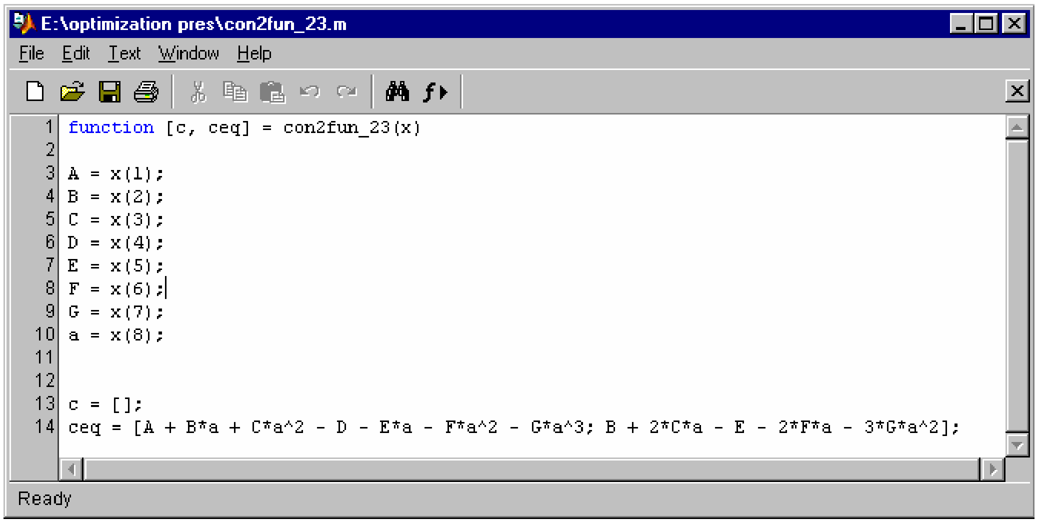
```matlab
global twoR
load twoR.dat
%Initialized X0
options = optimset('LargeScale','off');
[x, feval] = fmincon(@myfun_CON2_23,x0,[],[],[],[],[],[],@con2fun_23,options)

indata = twoR(:,1);
outdata = twoR(:,2);

% Vector x components;
A = x(1);
B = x(2);
C = x(3);
D = x(4);
E = x(5);
F = x(6);
G = x(7);
a = x(8);

indx1 = find(indata >= a);
y1 = A + B*indata(indx1) + C*indata(indx1).^2;

indx2 = find(indata <= a);
y2 = D + E*indata(indx2) + F*indata(indx2).^2 + G*indata(indx2).^3;

y = [y1;y2];

clf;
plot(indata(indx1),y1,'--r');
hold on;
axis([-20 60 0 25]);
plot(indata(indx2),y2,'k');
xlabel 'Electric field'
ylabel 'Strain'
title 'Figure.5b'
plot(indata,outdata,'s','markersize',3.5);

```

Objective Function Code



```
function f = myfun_CON2_23(x)

global twoR

indata = twoR(:,1);
outdata = twoR(:,2);

A = x(1);
B = x(2);
C = x(3);
D = x(4);
E = x(5);
F = x(6);
G = x(7);
a = x(8);

indx1 = find(indata <= a);
y1 = A + B*indata(indx1)+ C*indata(indx1).^2;

indx2 = find(indata >= a);
y2 = D + E*indata(indx2) + F*indata(indx2).^2 + G*indata(indx2).^3;

y = [y1;y2];

f = (sum((outdata-y).^2))^(1/2);

```

Constraints Code



```
function [c, ceq] = con2fun_23(x)

A = x(1);
B = x(2);
C = x(3);
D = x(4);
E = x(5);
F = x(6);
G = x(7);
a = x(8);


c = [];
ceq = [A + B*a + C*a^2 - D - E*a - F*a^2 - G*a^3; B + 2*C*a - E - 2*F*a - 3*G*a^2];
```