Last Name Ventura_____ **Name** _____Lara__ **matricola** ___________

PC No. ______1_____

Section 1: LINEAR PROGRAMMING

Objective function

minimize z = 22x + 26(n^4-0.25)/(n^4+0.25)y
(where n = PC number)
subject to

Constraints

```
100x +10y       >=200
-20x -10y       >= -120
20x  +50y       >=360
20x  +100y      >=400
x    +(n⁴-0.25)/(n⁴+0.25)y      <=25
```

Non-negativity constraints

x, y ≥ 0

Questions

a. Is the above LP problem in the **standard form**?

b. Find the value of the **objective function** and the corresponding values of the **decision variables** at each vertex

c. Find the **minimum** (optimal value of the **objective function** and optimal values of the **decision variables**), by briefly explaining the used procedure and tools

d. What happens if you turn the search into the **maximum**? Does it exist? How much is it?

e. What happens if you remove the last constraint: x+(n^4-0.25)/(n^4+0.25)y<=25?

f. What happens in the original problem if x becomes **unconstrained in sign**, that is $x \in \Re$? How can you handle this case?

Section 2: EMPIRICAL MODELS

Using Matlab® for the inactivation data in the attached file HHP-data_16-01-2015.xls measured during a pasteurization treatment of Orange juice by High Hydrostatic Pressure (HHP) treatment at different pressure levels:

a. determine the best **regression model**
b. report values for the **performance indexes** of the regression model
c. discuss **formulas and meaning** of the performance indexes
d. using the best **regression model,** estimate the Log reduction of the fruit juice after the application of a pressure of 350 MPa.

Solution

Section 1.

N=1;
The objective function is:

z = 22x -15.6 y min!

The constrains are:

100x	+10y	>=200
-20x	-10y	>= -120
20x	+50y	>=360
20x	+100y	>=400
x	+(3/5)y	<=25

is not in a standard form because to be in a standard form it must have:

1.Non-negativity for all variables (is already satisfied).In order to have a physical or chemical meaning, the decision variables must be zero or positive.

2.Non negativity for resourses (right hand sides of the equations) (we have to change to satisfy this requirement)

3.All the constraints must be in the form of equality. (for this requirement must be added slack and surplus variables).

After that we have:

z = 22x -15.6 y min!

The constrains are:

100x	+10y	>=200
20x	+10y	<= 120
20x	+50y	>=360
20x	+100y	>=400
x	+(3/5)y	<=25

We are working with two **decision variables** (x,y). It is a typical linear programming problem because both objective function and constraints are linear.

Moreover we have two decision variables, so we can use the graphical method. The **standard form** in vector-matrix representation is:

$$\text{Max } c^T x$$
$$Ax \geq b$$
$$x \geq 0$$

The objective function has to be minimized:

c^T is the cost coefficients row vector, $c^T = (22, 15.6)$;

the decision variables are in the number k = 2, x = (x, y);

A is an *m*n* matrix where *m* is the number of constrains (m=5) and *n* is the number of global variables:

A=

100.0000	10.0000
20.0000	10.0000
20.0000	50.0000
20.0000	100.0000
1.0000	0.6000

The matrix's element a_{ij} is called absorption coefficient.
b is a column vector are known terms:

b= 200

120

360

400

25

The graphical method helps us to solve the problem with a representation in a Cartesian plane. The straight line is plotted in place of each inequality. The different directions of the solution depend on inequalities' signs (> or <).
In this way we identify a particular domain that is called ***"feasible region"***, the set in the space of the decision variables of all points that satisfy all the constraints of the model at the same time. The optimal condition is found at the intersection of several constraints, and not in the interior of the convex region where the inequality constrains can be satisfied. Therefore we could move along constraints which improve the value of the objective function.
Generally the optimum for a linear programming problems always lies at a vertex of feasible region (Fundamental Theorem). The corner point with the best objective function value is optimal.
In Matlab we use the function ***drawfr (c, A, rel, b)*** in order to solve LP-2D problems.
drawfr (c, A, rel, b) where:
c = cost coefficient vector;
A = absorption cost matrix;
rel = variable that contains the signs of inequalities;
b = column vector.

When we have constraints of type = :

$a_{i1}\ x_1 + a_{i2}\ x_2 + ... + a_{ij}\ x_j + ... + a_{in}\ x_n\ = b_i$

it is necessary to modify in this way:

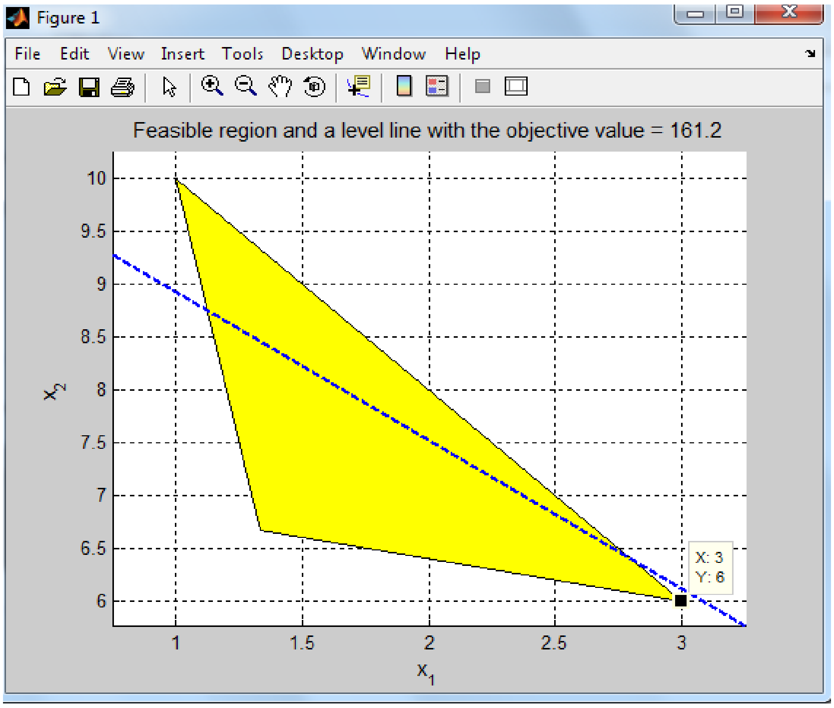
$a_{i1}\ x_1 + a_{i2}\ x_2 + ... + a_{ij}\ x_j + ... + a_{in}\ x_n\ \leq b_i'$

$a_{i1}\ x_1 + a_{i2}\ x_2 + ... + a_{ij}\ x_j + ... + a_{in}\ x_n\ \geq b_i''$

with $b_i' \approx< b_i \approx< b_i''$

In our case all constrains are inequalities.

The software Matlab show us this figure:



we can identify different vertexes,here are shown the coordinates:

x	**y**
1	10
1.33	6.667
3	6

a. In the first vertex the value of z=161.2

b. In the second vertex :133.3312

c. In the third :159.6

The feasible region is represented by the yellow figure and the basic feasible solutions are in the vertex of this region. The blue dashed line is the objective function. We move the objective function's line towards right if we want to maximize the function or towards left if we want to minimize the function. The coordinates of the vertex in which the objective function is minimized, are the optimal solution of my problem. In this case the optimal values of the decision variables are:

x=1.333

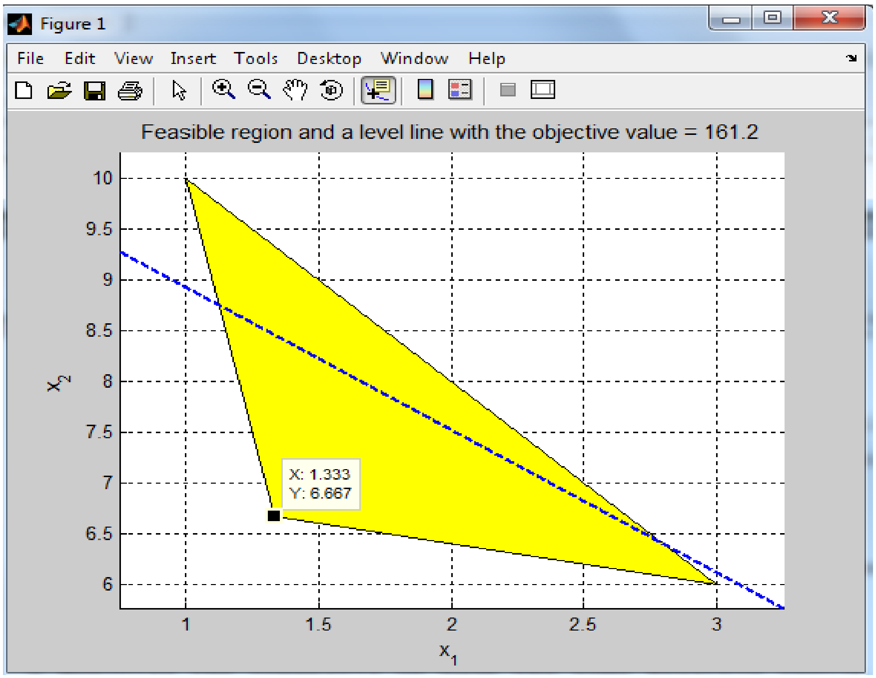
y=6.667 the value of the objective function is: 133.3312

d.To find the maximum we must move in the opposite direction after that we have that in this case the optimal value for the decision variables is:

x=1

y=10 and the value of the objective function is :159.6 then we have this solution and we can individuate one optimal solution.

e. Removing the constraint, we have the same feasible region and, since the objective function is the same, nothing changes.



f. Since to work is important that all the constrains are respected then that x is positive this variable belong to $\Re$ then to work we can substitute x with a new variable defined in this way like: x1=-x and in this way if x is negative replacing it with x1 we can normally work.

Section 2

Using Matlab and changing the directory with Import Data we obtain a matrix data of (6*2) ,6 rows and 2 columns,after that we define the variables to plot them:

x=data(:,1);

y=data(:,2);

x = (independent variable)

0

50

100

150

200

250

y=(dependent variable)

y =

0

-0.3699

-0.6745

-1.1387

-1.9489

-3.4859

The empirical model can be developed in two different approaches:

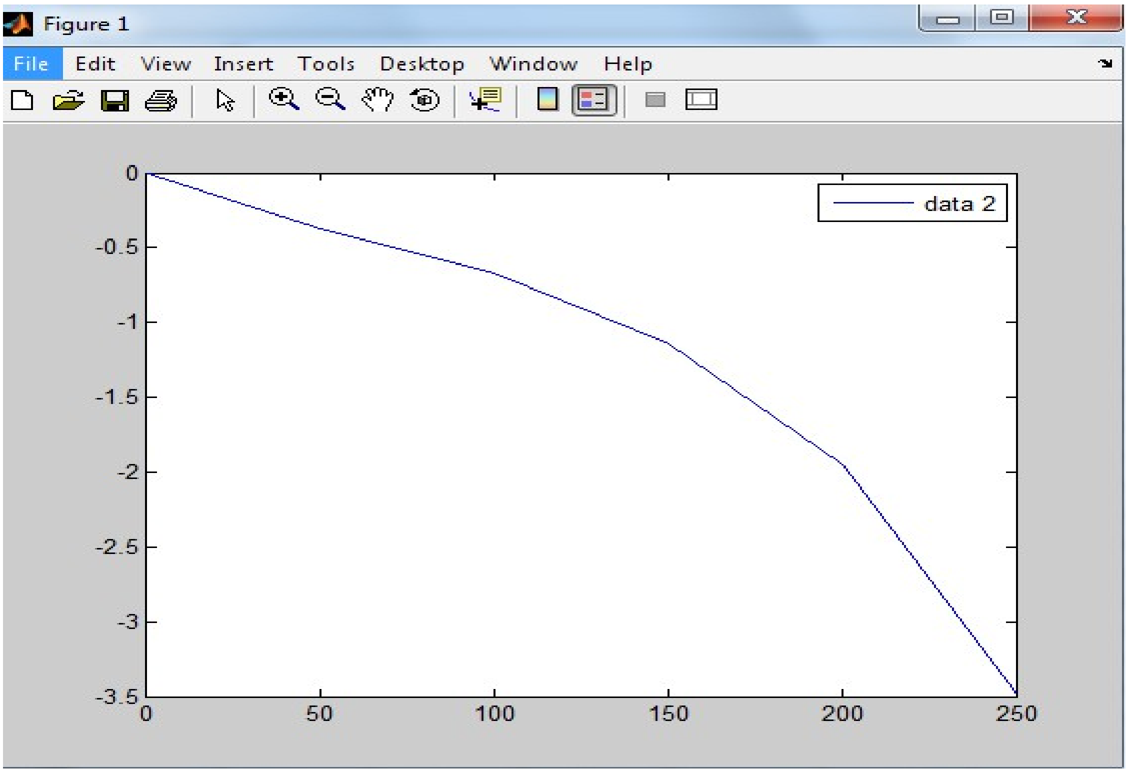
Interpolation: in this case we construct a curve (often polynomial) that contains all the data points. It is also possible to make an interpolation by *spline,* that is a piece-wise polynomial function: we divide the entire interval of interpolation into *n* smaller sub-intervals and consider a different polynomial for each of these sub-intervals (same degree but different coefficients).

a. **Regression or smoothing**: we construct a curve that is close as possible to containing all the data points.

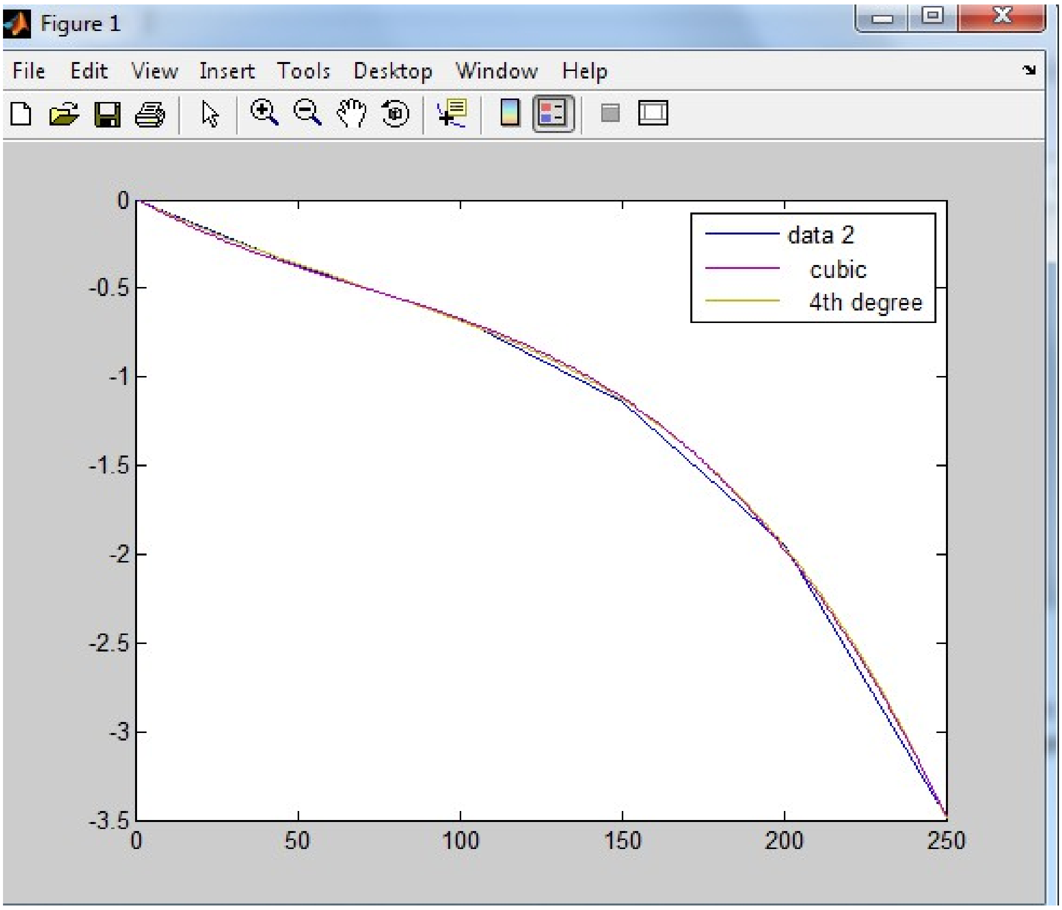
Matlab supports curve fitting through the Basic Fitting Interface or the Curve Fitting Toolbox.

BASIC FITTING INTERFACE

The first thing that we have to do is to import the data from the *current directory* into Matlab. We open the window *Import Data* and then we create a matrix saved in the *workspace*. After we duplicate the matrix in order to generate two column vectors; each vector has got one column, the first one contains the variable x and the second one the variable y. At the end we plot the data:



In this figure we have a line that represents the data point in a Cartesian plane. If we select *Basic Fitting* from the *Tools menu*, we can choose some interpolation (for example spline) and regression (linear, quadratic, cubic polynomial) models.
Testing with cubic and 4 th degree polynomial we have:



cubic:
y = p1*x^3 + p2*x^2 + p3*x + p4

Coefficients:
p1 = -3.3485e-007
p2 = 6.944e-005
p3 = -0.010368
p4 = 0.0046502

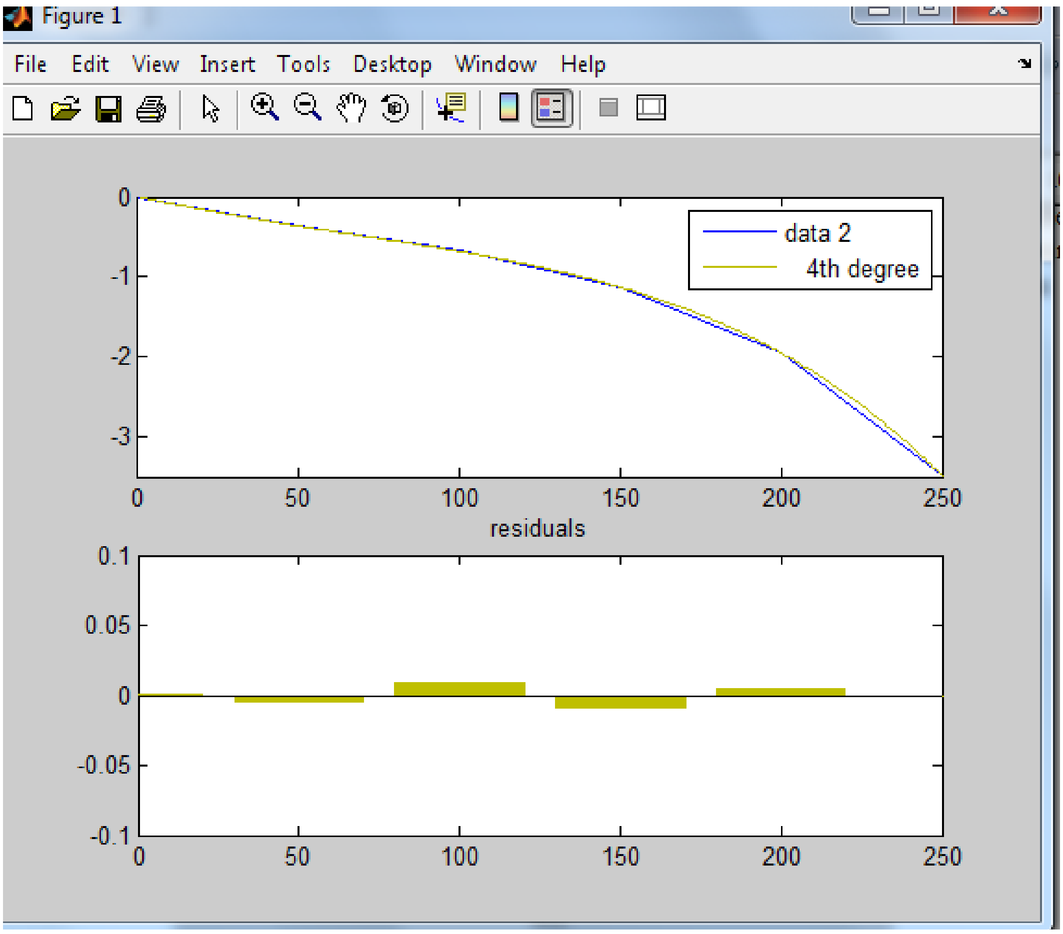
Norm of residuals =
0.032947

4^{th} polynomial:
y = p1*x^3 + p2*x^2 + p3*x + p4

Coefficients:
p1 = -3.3485e-007
p2 = 6.944e-005
p3 = -0.010368
p4 = 0.0046502

Norm of residuals =
0.032947

If we consider the residuals showed after they are very close to the zero's line in the last case:



CURVE FITTING TOOL

After the data loading, these data appear in the workspace and then we open the Curve Fitting Tool with the *cftool* command in the *command window*. Moreover we can make the **smoothing** (elimination of the "noise" from data). In general for each abscissa we can have two different ordinates and in this case it is necessary to do the mean. This technique allows to obtain a better fitting model with less oscillation, but is not necessary in this case.
From the Curve Fitting Tool window, we click on fitting and we can choose more different interpolation and regression models than the previous case we choose again a cubic function and we have:

Linear model Poly3:
    f(x) = p1*x^3 + p2*x^2 + p3*x + p4
Coefficients (with 95% confidence bounds):
    p1 = -3.349e-007 (-4.345e-007, -2.352e-007)
    p2 = 6.944e-005 (3.151e-005, 0.0001074)
    p3 =   -0.01037 (-0.01418, -0.006552)
    p4 =   0.00465 (-0.09358, 0.1029)

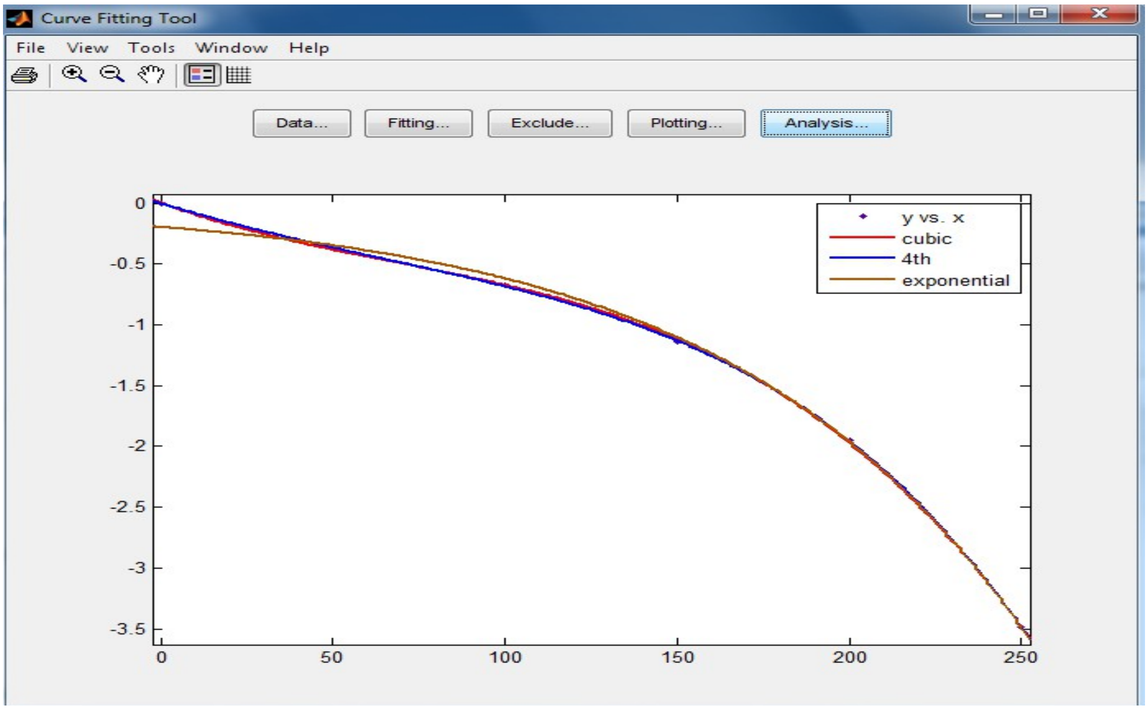
Goodness of fit:
  SSE: 0.001085
  R-square: 0.9999
  Adjusted R-square: 0.9997
  RMSE: 0.0233

Linear model Poly4:
    f(x) = p1*x^4 + p2*x^3 + p3*x^2 + p4*x + p5
Coefficients (with 95% confidence bounds):
    p1 = -5.203e-010 (-3.809e-009, 2.768e-009)
    p2 = -7.47e-008 (-1.73e-006, 1.58e-006)
    p3 = 2.949e-005 (-0.0002327, 0.0002917)
    p4 =   -0.00851 (-0.02223, 0.005214)
    p5 = -0.0009244 (-0.187, 0.1852)

Goodness of fit:
  SSE: 0.0002153
  R-square: 1
  Adjusted R-square: 0.9999
  RMSE: 0.01467

General model Exp1:
    f(x) = a*exp(b*x)
Coefficients (with 95% confidence bounds):
    a =   -0.1953 (-0.2726, -0.1181)
    b =   0.01153 (0.00982, 0.01325)

Goodness of fit:



SSE: 0.04328
R-square: 0.9947
Adjusted R-square: 0.9934
RMSE: 0.104

Then the best result is obtained with the linear model Poly4.

b. index values:
Goodness of fit:
SSE: 0.0002153
R-square: 1
Adjusted R-square: 0.9999
RMSE: 0.01467

PERFORMANCE INDEXES OF THE REGRESSION MODEL
The accuracy or the goodness of fit is expressed by several indexes.
The **Sum of Squared Errors SSE** (or S_r) is the sum of residuals' squares:

$$SSE = \sum_{i}^{n} [(yi - f(xi)]^2$$

We have a better fitting when SSE value is near to zero.
The **determination coefficient R^2** is calculated in this way:

$$R^2 = \frac{St - Sr}{St}$$

where *St* is the Sum of Squares about the Mean:

$$St = SST = \sum_{i}^{n} (yi - ym)^2$$

in which y^m is the arithmetic mean:

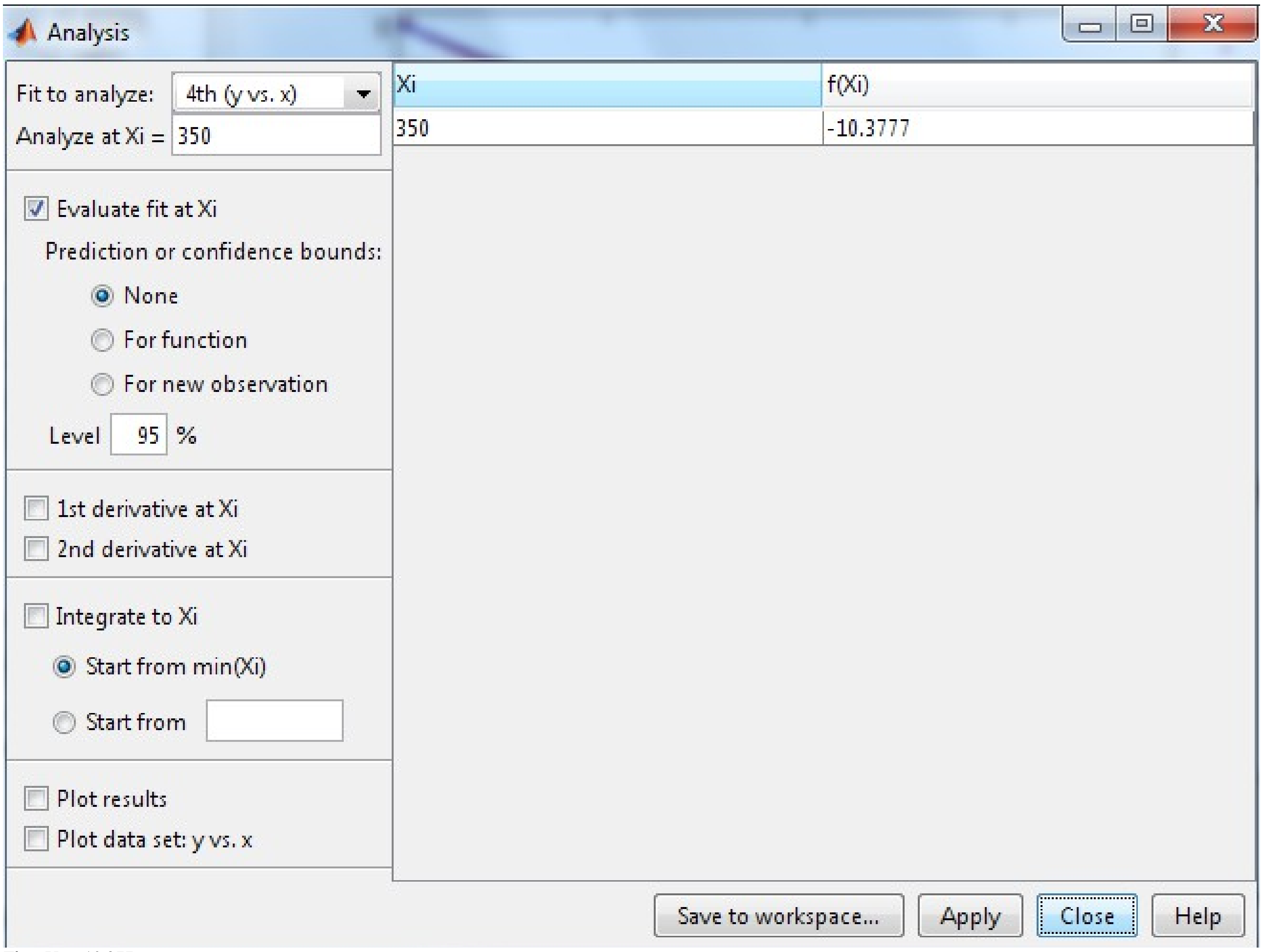
$$ym = \frac{\sum yi}{n}$$

R^2 is an important index in order to compare different functions and after this comparison, we choose the function having R^2 near to 1 because it reduces the gap between experimental data and model fitting values.
The **adjusted r-square (adj r^2)** is the R^2 corrected according to the independent variable number and it is the best index when we add some coefficients to the model.
The **Root Mean Squared Error** is defined as:

$$RMSE = \sqrt{\left(\frac{\sum_i^n (y_i - f(x_i))^2}{n - n_{parametri}}\right)}$$

where n is the number of data points and $n_{parametric}$ is the number of parameters in the particular model. The difference (n-$n_{parametric}$) is the number of degrees of freedom. RMSE has to be near to zero because indicates a fit that is more useful for prediction.
In this case the index parameter that we have interested is R^2.

d. Using Analysis in cftool we can obtain after the chosen of the 4th polynomial fit the value of the Log reduction it results like this:



Then Y = -10.377