1. **Mathematical background**

i. Linear Regression

Using minima squares is possible to compute a straight line that approximate in the best way all point given, where that line is described by $y = mx + b$. That before, for at least three pair of points.

To compute the values of m and b we follow the next equations (deductions left to the reader):

$$m = \frac{N \sum_{i=1}^{N} x_i y_i - \left(\sum_{i=1}^{N} x_i \right) \left(\sum_{i=1}^{N} y_i \right)}{N \sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2} \tag{1}$$

$$b = \frac{\left(\sum_{i=1}^{N} y_i \right) \left(N \sum_{i=1}^{N} x_i^2 \right) - \left(\sum_{i=1}^{N} x_i \right) \left(\sum_{i=1}^{N} x_i y_i \right)}{N \sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2} \tag{2}$$

How the line approximate the numbers given is measured by the factor r, and when its is equal to 1 all values are described by the line. That factor can be computed by:

$$r = \frac{\sum_{i=1}^{N} x_i y_i - N^{-1} \left(\sum_{i=1}^{N} x_i \right) \left(\sum_{i=1}^{N} y_i \right)}{\sqrt{\left[\sum_{i=1}^{N} x_i^2 - N^{-1} \left(\sum_{i=1}^{N} x_i \right)^2 \right] \left[\sum_{i=1}^{N} y_i^2 - N^{-1} \left(\sum_{i=1}^{N} y_i \right)^2 \right]}} \tag{3}$$

When the r is not 1 (majority of cases) is useful know the uncertainties for m and b, which can be computed by:

$$u_m = \sqrt{\frac{N \sum_{i=1}^{N} (y_i - m x_i - b)^2}{(N-2) \left[N \sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2 \right]}} \tag{4}$$

$$u_b = \sqrt{\frac{\left[\sum_{i=1}^{N} (y_i - m x_i - b)^2 \right] \left[\sum_{i=1}^{N} x_i^2 \right]}{(N-2) \left[N \sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2 \right]}} \tag{5}$$

ii. Polynomial Regression

It is also know that given $n+1$ pair points is possible to compute a polynomial curve at n^{tm} grade that satisfies all the pairs. Also is computable a $n-m$ polynomial regression with $m<n$, where the approach of the points by the curve can be modified in way of the grade polynomial.

Given n pairs: $\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$ a polynomial regression can be computed as: $y = a_n x^n + a_{n-1}x^{n-1} + ... + a_2 x^2 + a_1 x + a_0$
where the coefficients are determined by the following system of linear equations.

$$\begin{array}{ccccccccc}
na_0 & + & (\sum x_i)a_1 & + & (\sum x_i^2)a_2 & +...+ & (\sum x_i^n)a_n & = & \sum y_i \\
(\sum x_i)na_0 & + & (\sum x_i^2)a_1 & + & (\sum x_i^3)a_2 & +...+ & (\sum x_i^{n+1})a_n & = & \sum x_i y_i \\
(\sum x_i^2)na_0 & + & (\sum x_i^3)a_1 & + & (\sum x_i^4)a_2 & +...+ & (\sum x_i^{n+2})a_n & = & \sum x_i^2 y_i \\
 & & & & & & & \vdots & \\
(\sum x_i^n)na_0 & + & (\sum x_i^{n+1})a_1 & + & (\sum x_i^{n+2})a_2 & +...+ & (\sum x_i^{2n})a_n & = & \sum x_i^n y_i
\end{array} \tag{6}$$

Therefore, is possible to solve polynomial regression by solving the linear system.

2. Code

The two cases described above are in two different files (`linear.f90` and `polynomial.f90`). The first one about a linear regression was easy to code since all expression are fixed. Doing the additions by `DO` loops and the square root is a native function in `fortran` was not necessary import any library.

For the second case, about the polynomial regression the code is little bit more complicated, since is necessary not only a solver for the linear equation, is also necessary compute the values for the linear system. Then, was written two `SUBROUTINES`, one for compute the values and another one to solve the system.

Before compute, there are a conditional to know if the system can be solved or not, if not the program stops and gives a suggestion for a better try.

3. Examples

We can do a polynomial regression with the next pairs: *i*) (1,1), (2,4), (3,4), (5,1), (6,2), and *ii*) (-3,4), (-1,1), (0,0), (1,2), (2,5). At 3th grade we have:

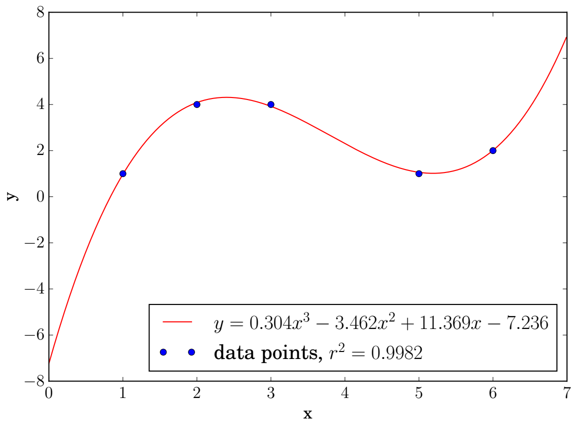

Figure 1: Example *i*.

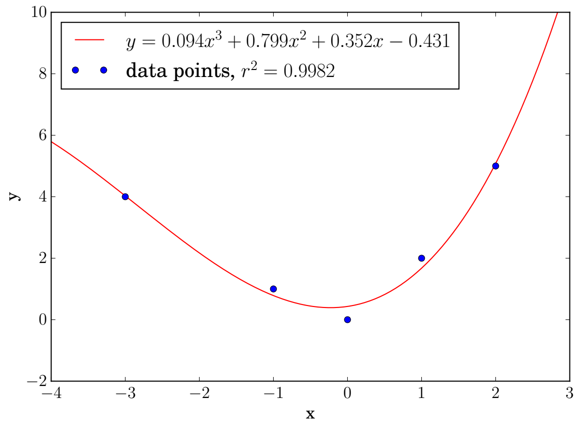

Figure 2: Example *ii*.

Also we can compute for: $(-\pi/2,\, 0)$, $(-\pi/3,\, 1/2)$, $(0,\, 1)$, $(\pi/3,\, 1/2)$, $(\pi/2,\, 0)$. And compare with the function $cos(\pi/4)$. Getting a approximation of the Taylor expansion for $cos(x)$.

$$\cos(x) = \sum_{n=0}^{\infty}(-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots \approx 0.966 - 0.397x^2$$

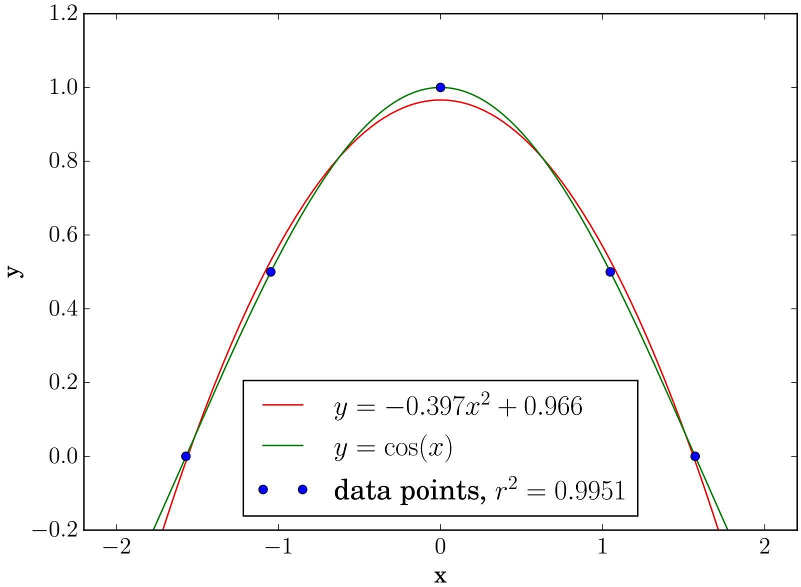

Figure 3: Example *iii*.