PolyFit: A C++ code for polynomial curve fit with calculation of error bars

Ianik Plante[1]

[1]KBR, 2400 NASA Parkway, Houston 77058
Ianik.Plante-1@nasa.gov

Abstract

In radiobiology, many dose-response results are modeled using the so-called linear-quadratic (LQ) model, which means that results are modeled as a function of dose D as $R(D) = \beta_0 + \beta_1 D + \beta_2 D^2$. The coefficients β_0, β_1 and β_2 are obtained from fitting a series of data points (x_i, y_i), which is usually done using a least-square method. The LQ and more generally the polynomial fit capability is implemented in many software that analyzes data. However, it is often convenient to do the fitting programmatically, especially when a large number of datasets should be analyzed. Furthermore, depending on the software used, some features may not be implemented. In this mini-review, I discuss the basis of polynomial fitting, including the calculation of errors on the coefficients and results, use of weighting and fixing the intercept value (the coefficient β_0). A simple C++ code to perform the polynomial curve fitting is also provided. This code should be useful not only in radiobiology but in other fields of science as well.

1. Introduction

For a given dataset (x_i, y_i), *i* = 1,2, ..., *n*, where *x* is the independent variable and *y* is the dependent variable, a polynomial regression fits data to a model of the following form:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k + \varepsilon_i = \sum_{j=0}^{k} \beta_j x_i^j + \varepsilon_i \tag{1}$$

where *k* is the polynomial order. In general, *k* is a small integer number. The parameters β_k are estimated using a weighted least-square method. This method minimizes the sum of the squares of the deviations between the theoretical curve and the experimental points for a range of independent variables (Chernov, 2010).

The quantity β_0 is the y-intercept and the parameters $\beta_1, \beta_2, \ldots, \beta_k$ are the "partial coefficients" (or "partial slopes"). The set of equations (1) can be written conveniently in matrix form:

$$Y = XB + E, \tag{2}$$

where

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}; X = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix}; B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}; \mathrm{E} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{3}$$

Y is a *n×1* column vector, *X* is a *n×(k+1)* matrix, *B* is a *k×1* column vector, *E* is a *n×1* column vector. Furthermore, ε_i are distributed as normal random variables with $\bar{\mathrm{E}} = 0$ and $Var(\mathrm{E}) = \sigma^2$.

2. Calculation of the coefficients $\widehat{\boldsymbol{\beta}}_k$

To calculate the coefficients B that minimize the error $\|\mathrm{E}\|^2$, the derivates with respect to B are calculated and set equal to 0:

$$\frac{\partial\|\mathrm{E}\|^2}{\partial\beta_m} = 0, \tag{4}$$

where $m = 0, \ldots, k$. This can be written explicitly as

$$\|\mathrm{E}\|^2 = \sum_{i=1}^{n}\varepsilon_i^2 = \sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{k}\beta_j x_i^j\right)^2, \tag{5a}$$

$$\|\mathrm{E}\|^2 = \sum_{i=1}^{n}\left(y_i^2 - 2y_i\sum_{j=0}^{k}\beta_j x_i^j + \sum_{j=0}^{k}\sum_{l=0}^{k}\beta_j\beta_l x_i^{j+l}\right), \tag{5b}$$

So that

$$\frac{\partial\|\mathrm{E}\|^2}{\partial\beta_m} = \sum_{i=1}^{n}\left(-2y_i\sum_{j=0}^{k}x_i^j\,\delta_{jm} + \sum_{j=0}^{k}\sum_{l=0}^{k}\delta_{jm}\beta_l x_i^{j+l} + \sum_{j=0}^{k}\sum_{l=0}^{k}\beta_j\delta_{lm}x_i^{j+l}\right), \tag{6}$$

where δ_{ij} is the Kronecker delta. This simplifies to

$$\frac{\partial\|\mathrm{E}\|^2}{\partial\beta_m} = \sum_{i=1}^{n}\left(-2y_i x_i^m + \sum_{l=0}^{k}\beta_l\, x_i^{m+l} + \sum_{j=0}^{k}\beta_j x_i^{j+m}\right). \tag{7}$$

Changing summation indices simplifies the equation further:

$$\frac{\partial\|\mathrm{E}\|^2}{\partial\beta_m} = \sum_{i=1}^{n}\left(-2y_i x_i^m + 2\sum_{j=0}^{k}\beta_j x_i^{j+m}\right). \tag{8}$$

Equating to 0, the following equations are obtained ($m = 0, \ldots, k$):

$$\sum_{i=1}^{n}y_i x_i^m = \sum_{i=1}^{n}\sum_{j=0}^{k}\beta_j x_i^{j+m}. \tag{9}$$

These equations can also be written in matrix form as

$$\begin{bmatrix}\sum y_i\\ \sum x_i y_i\\ \vdots\\ \sum x_i^k\end{bmatrix} = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^k\\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{k+1}\\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{k+2}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \sum x_i^k & \sum x_i^{k+1} & \sum x_i^{k+2} & \cdots & \sum x_i^{k+k}\end{bmatrix}\begin{bmatrix}\beta_0\\ \beta_1\\ \vdots\\ \beta_k\end{bmatrix}, \tag{10}$$

where all sums runs from *i=1* to *n*. This can be further expressed with the matrices *X*, *Y* and *B* defined earlier:

$$\begin{bmatrix}1 & 1 & 1 & \cdots & 1\\ x_1 & x_2 & x_3 & \cdots & x_n\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ x_1^k & x_2^k & x_3^k & \cdots & x_n^k\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{bmatrix} = \begin{bmatrix}1 & 1 & 1 & \cdots & 1\\ x_1 & x_2 & x_3 & \cdots & x_n\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ x_1^k & x_2^k & x_3^k & \cdots & x_n^k\end{bmatrix}\begin{bmatrix}1 & x_1 & x_1^2 & \cdots & x_1^k\\ 1 & x_2 & x_2^2 & \cdots & x_2^k\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & x_n & x_n^2 & \cdots & x_n^k\end{bmatrix}\begin{bmatrix}\beta_0\\ \beta_1\\ \vdots\\ \beta_k\end{bmatrix}, \tag{11}$$

$$X^T Y = X^T XB, \tag{12}$$

where X^T is the transpose of X. Therefore B can be expressed in matrix form as

$$B = (X^T X)^{-1} X^T Y. \tag{13}$$

The result $\hat{B}$ is the **least square estimate** of the vector B, and it is the solution to the linear equations, which can be written as:

$$\hat{B} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} = (X^T X)^{-1} X^T Y, \tag{14}$$

The predicted value of Y for a given X is:

$$\hat{Y} = X\hat{B}, \tag{15}$$

By substituting $\hat{B}$ into (15), we define the matrix H as:

$$\hat{Y} = [X(X^T X)^{-1} X^T]Y = HY, \tag{16}$$

Note these important properties of the matrix H:

$$H^T = [X(X^T X)^{-1} X^T]^T = (X^T)^T [(X^T X)^{-1}]^T X^T = X[(X^T X)^T]^{-1} X^T = H, \tag{17a}$$

$$H^2 = [X(X^T X)^{-1} X^T][X(X^T X)^{-1} X^T] = H, \tag{17b}$$

So that H is an idempotent matrix, i.e. $H^2 = H = H^T$.

3. The residual sum of squares

The residuals are defined as:

$$res_i = y_i - \hat{y}_i, \tag{18}$$

and the residual sum of squares (RSS) can be written by:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \|\mathrm{E}\|^2, \tag{19}$$

The RSS can also be written using

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \left(Y - \hat{Y}\right)^T \left(Y - \hat{Y}\right) = (Y - HY)^T (Y - HY), \tag{20a}$$

$$RSS = Y^T \left(I_n - H^T\right)(I_n - H)Y = Y^T \left(I_n - H - H^T + H^T H\right)Y = Y^T (I_n - H)Y, \tag{20b}$$

Where I_n is an identity matrix with n elements. Since $H^2 = H$, it can be shown that the eigenvalues of this matrix are either 0 or 1:

$$Hv = lv; \tag{21a}$$

$$H^2 v = H(Hv) = H(lv) = lHv = l^2 v \tag{21b}$$

So that $l^2 = l$. This implies that $l = 0$ or $l = 1$. Furthermore, the sum of the eigenvalues equals the trace of the matrix, so that

$$Tr(I_n - H) = Tr(I_n) - Tr(H) = n - Tr(X(X^T X)^{-1} X^T) \tag{22a}$$

$$Tr(I_n - H) = n - Tr\big((X^T X)(X^T X)^{-1}\big) = n - (k + 1) \tag{22b}$$

In the last equation, the invariance property of the trace operator over cyclic permutation was used. Specifically, $Tr(ABC) = Tr(CAB)$.

Since *H* has *n* eigenvalues, all equal to 1 or 0, and since their sum is equal to *n-k-1*, then *n-k-1* must be equal to *1*, and *k+1* equal to *0*. This can be used to obtain the spectral decomposition of the matrix *I-H*:

$$I - H = ADA^T; \tag{23}$$

The matrix *D* can be written as

$$D = \begin{pmatrix} I_{n-k-1} & 0_{[n-k-1][k+1]} \\ 0_{[k+1][n-k-1]} & 0_{[k+1][k+1]} \end{pmatrix}; \tag{24}$$

Since *I-H* is symmetric, *A* is orthogonal, i.e. $A^T A = AA^T = I$. Since

$$HX = X \Rightarrow (I - H)X = 0 \Rightarrow ADA^T X = 0 \Longrightarrow DA^T X; \tag{25}$$

Hence

$$(A^T X)_{ij} = 0 \text{ for } i=1,\dots,n\text{-}k\text{-}1 \text{ and } j=1,\dots,n\text{-}k\text{-}1. \tag{26}$$

So that

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = Y^T ADA^T Y = \sum_{i=1}^{n-k-1} (A^T Y)_i^2, \tag{27}$$

Now, since $Y \sim N(X\beta, \sigma^2 I)$, then $A^T Y \sim N(A^T X\beta, \sigma^2 A^T A) = N(A^T X\beta, \sigma^2 I)$, so that the components of $A^T Y$ are independent. Since the sum of the square of *p* independent normal variates of variance σ^2 is a chi-square distribution with p degrees of freedom, than the RSS is distributed as chi-square distribution with *n−k−1* degrees of freedom. From this σ^2 can be calculated as

$$\sigma^2 = \frac{RSS}{n-k-1} \tag{28}$$

4. Calculation of the standard error of coefficients $\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}$

To calculate the error, first calculate the expected value of $\hat{\beta}_k$, $E(\hat{B})$. Using equation 15, we get:

$$E(\hat{B}) = E[(X^T X)^{-1} X^T Y]. \tag{29}$$

Since $(X^T X)^{-1} X^T$ are fixed, they are considered constants, so that

$$E(\hat{B}) = (X^T X)^{-1} X^T E[Y]. \tag{30}$$

Now we can use equation 2:

$$E(\hat{B}) = (X^T X)^{-1} X^T E(XB + E), \tag{31a}$$

$$E(\hat{B}) = (X^T X)^{-1} X^T E(XB) + (X^T X)^{-1} X^T E(\mathrm{E}), \tag{31b}$$

Since $E(XB) = XB$ and $E(\mathrm{E}) = 0$, the equation simplifies to:

$$E(\hat{B}) = (X^T X)^{-1} X^T XB + (X^T X)^{-1} X^T 0 = B, \tag{32}$$

To calculate the variance, for a matrix A and a vector y, it is known that $Var(Ay) = AVar(y)A^T$. Hence

$$Var(\hat{B}) = Var((X^T X)^{-1} X^T Y), \tag{33a}$$

$$Var(\hat{B}) = [(X^T X)^{-1} X^T] Var(Y) [(X^T X)^{-1} X^T]^T, \tag{33b}$$

Since $Y = XB + E$, $Var(Y) = \sigma^2 I$, where I is the identity matrix. Hence

$$Var(\hat{B}) = [(X^T X)^{-1} X^T] \sigma^2 I X [(X^T X)^{-1}]^T, \tag{34a}$$

$$Var(\hat{B}) = \sigma^2 (X^T X)^{-1} (X^T X) [(X^T X)^{-1}]^T, \tag{34b}$$

$$Var(\hat{B}) = \sigma^2 [(X^T X)^T]^{-1} = \sigma^2 (X^T X)^{-1}, \tag{34c}$$

The standard errors on coefficients are therefore

$$S_{\widehat{\beta_J}} = \sigma \sqrt{(X^T X)^{-1}_{jj}} = \sqrt{\frac{RSS}{n-k-1} (X^T X)^{-1}_{jj}}, \tag{35}$$

The matrix $\sigma^2 (X^T X)^{-1}$ is the covariance matrix.

5. Confidence interval of parameters

The *t*-values of the coefficients can be computed as:

$$t = \frac{\beta_j - 0}{S_{\widehat{\beta_J}}}, \tag{36}$$

From the *t*-value, the $(1 - \alpha) \times 100\%$ **Confidence Interval** for each parameter can be calculated by:

$$\widehat{\beta_J} - t_{\left(\frac{\alpha}{2}, n-k-1\right)} S_{\widehat{\beta_J}} \leq \widehat{\beta_J} \leq \widehat{\beta_J} + t_{\left(\frac{\alpha}{2}, n-k-1\right)} S_{\widehat{\beta_J}}, \tag{37}$$

If the regression assumptions hold, we can perform the t-tests for the regression coefficients with the null hypotheses and the alternative hypotheses:

$$H_0 : \beta_j = 0, \tag{38a}$$

$$H_1 : \beta_j \neq 0, \tag{38b}$$

With the *t*-value, we can decide whether to reject the corresponding null hypothesis. Usually, for a given **Confidence Level for Parameters**: α, we can reject H_0 when $|t| > t_{\alpha/2}$. Additionally, the *p*-value is less than α.

Prob>|t|

This is the probability that H_0 in the t test is true, which is calculated as

$$prob = 2(1 - tcdf(|t|, df_{Error})), \tag{39}$$

where $tcdf(|t|, df_{Error})$ is the cumulative distribution function of the Student's t distribution at the values |t|, with **degree of freedom of error** $df_{Error} = n - k - 1$.

6. Calculation of prediction and confidence bands

The confidence interval for the fitting function says how good the estimate of the value of the fitting function is at particular values of the independent variables. In other words, the correct values for the fitting function lies within the confidence interval with confidence level 100α%, which is given by

$$\hat{y} \pm t_{\frac{\alpha}{2}, n-k-1} \sigma\left(X^{*T} (X^T X)^{-1} X^*\right), \tag{40}$$

where

$$X^* = \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^k \end{bmatrix} \tag{41}$$

is a *(k+1)×1* column vector calculated at a given *x* value. Similarly, the prediction interval for the confidence level is the interval within which 100 of all experimental data points in a series of repeated measurements are expected to fall at particular values of the independent variables. This is given by

$$\hat{y} \pm t_{\frac{\alpha}{2}, n-k-1} \sigma\left(1 + X^{*T} (X^T X)^{-1} X^*\right). \tag{42}$$

7. Weighted fitting

In some cases, it is convenient to use weighted fitting. The weight of each point is set to 1 by default. Usually, the weights are given by $w_i = \sigma_i$ or $w_i = 1/\sigma_i^2$, where is the error on the point σ_i. The weight matrix *W* is therefore

$$W = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix} \tag{43}$$

The matrix $X^T X$ is replaced by $X^T W X$ in most equations. The RSS is now given by

$$||\mathrm{E}||^2 = \sum_{i=1}^{n} w_i \varepsilon_i^2 = \sum_{i=1}^{n} w_i \left(y_i - \sum_{j=0}^{k} \beta_j x_i^j\right)^2, \tag{44}$$

The coefficients are given by

$$\hat{B} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} = (X^T W X)^{-1} X^T W Y, \tag{45}$$

7. Fix Intercept (at)

Fix intercept will set the y-intercept β_0 to a fixed value. In this case, the total degree of freedom will be *n*=n* due to the intercept fixed. The matrix *X* and *B* are changed to

$$X = \begin{bmatrix} x_1 & x_1^2 & \cdots & x_1^k \\ x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_n^2 & \cdots & x_n^k \end{bmatrix}; B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix}; \tag{46}$$

i.e. *X* of dimension *n×k*, and *B* is a *k×1* dimension column vector. Prior solving the system of equations, the values y_i should be translated by the desired fixed intercept β_0.

8. Correlation matrix

The correlation matrix is calculated using the elements $Cov(\beta_i, \beta_j)$ of the covariance matrix $\sigma^2(X^T W X)^{-1}$. The elements ρ_{ij} are calculated as follows:

$$\rho_{ij} = \frac{Cov(\beta_i, \beta_j)}{\sqrt{Cov(\beta_i, \beta_i)}\sqrt{Cov(\beta_j, \beta_j)}}, \tag{47}$$

9. The C++ code

The C++ code can be found at https://github.com/nasa/polyfit. It can be compiled in Linux using the command

g++ -o PolyFit PolyFit.cpp

The C++ code has been tested on Linux, but since it is written in basic C++, it can be compiled on other platforms.

The main subroutine requires the input values to be suitable, and the code performs only minimal validation. If a weighted fit is used, the weight values for all points should be greater than 0. It is also recommended to order the data points by increasing x values.

The calculation of critical values for the student-t test and the F test (ANOVA) requires the evaluation of special functions, which is beyond the scope of this text. Similarly, the calculation of the inverse of a matrix is a basic problem in linear algebra.

10. Example

The following example was done with the program, and compared to the fitting provided by the Origin® software.

X	Y	Y error
0	0	0.1
0.5	0.21723	0.3
1	0.43445	0.2
2	0.99924	0.4
4	2.43292	0.1
6	4.77895	0.3

Fitting parameters: Polynomial degree: 2. Intercept not fixed. Error weighted as $w_i = 1/\sigma_i^2$. The results are:

Param	Value		Standard error		t-value		Prob>|t|	
	Polyfit	Origin	Polyfit	Origin	Polyfit	Origin	Polyfit	Origin
β_0	0.0173268	0.01733	0.0315352	0.03154	0.549445	0.54944	0.620957	0.62096
β_1	0.261372	0.26137	0.0406847	0.04068	6.42433	6.42433	0.00764445	0.00764
β_2	0.0868543	0.08685	0.00850808	0.00851	10.2085	10.20845	0.00200349	0.002

Statistics

	Polyfit	Origin
Number of Points	6	6
Degrees of Freedom	3	3
Residual Sum of Squares	0.339429	0.33943
R-Square (COD)	0.999268	0.99927
Adj. R-Square	0.998779	0.99878

ANOVA (Polyfit)

	DF	Sum squares	Mean Square	F value	Prob >F
Model	2	463.082	231.541	2046.44	1.98483e-05
Error	3	0.339429	0.113143		
Total	5	463.421			

ANOVA (Origin)

	DF	Sum squares	Mean Square	F value	Prob >F
Model	2	463.08155	231.54077	2046.44269	1.98226E-5
Error	3	0.33943	0.11314		
Total	5	463.42097			

Covariance matrix (Polyfit)

	β_0	β_1	β_2
β_0	0.000994467	-0.00062013	8.63062e-05
β_1	-0.00062013	0.00165525	-0.00033444
β_2	8.63062e-05	-0.00033444	7.23874e-05

Covariance matrix (Origin)

	β_0	β_1	β_2
β_0	9.94467E-4	-6.2013E-4	8.63062E-5
β_1	-6.2013E-4	0.00166	-3.3444E-4
β_2	8.63062E-5	-3.3444E-4	7.23874E-5

Correlation matrix (Polyfit)

	β_0	β_1	β_2
β_0	1	-0.483344	0.321674
β_1	-0.483344	1	-0.966174
β_2	0.321674	-0.966174	1

Correlation matrix (Origin)

	β_0	β_1	β_2
β_0	1	-0.48334	0.32167
β_1	-0.48334	1	-0.96617
β_2	0.32167	-0.96617	1

Prediction and confidence bands

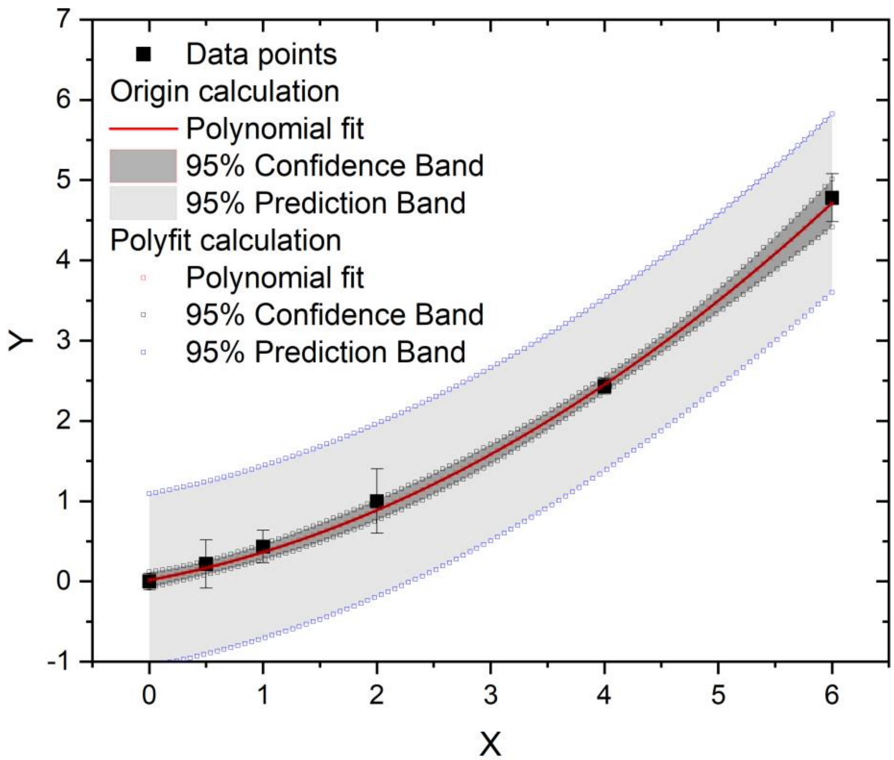

Figure 1. Polynomial fit, 95% confidence and 95% prediction bands as calculated by Polyfit (points) and Origin® (lines). The two calculations are indistinguishable.

11. Conclusion

We have reviewed the calculation of a polynomial fit of data, and relevant calculations such as the standard error and confidence intervals on the coefficients, correlation matrix, covariance matrix, and the 95% prediction and confidence bands. Several cases have been considered: fixed or variable intercept, and weighted coefficients. One case has been evaluated using the program, and compared with the results obtained by Origin®. In general, all calculations performed by Polyfit are identical to those made using Origin®, with small differences attributable to the precision limit.

References

1. N. Chernov (2010), *Circular and linear regression: Fitting circles and lines by least squares*, Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117 (256 pp.)

2. Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P. (2007). Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.

Acknowledgements

This work was supported by NASA Space Radiation Program (NNX16AR97G for MH), NASA Human Health and Performance Contract (HHPC) number NNJ15HK11B.