Analele Universităţii de Vest din Timişoara
Vol. LIV, 2010
Seria Fizică

INTERPOLATION WITH MATHCAD: A METHOD TO FIND PHYSICAL LAWS FROM EXPERIMENTAL DATA

C. D. Buioca

University of Petrosani, 20 University Street, 332006 Petrosani, Romania

Abstract
The paper presents a method for experimental data interpolation using the software MathCAD. It is written and explained a program allowing the processing of real experimental data, affected by experimental errors, obtained during lab and research activities. Some advantages of interpolation with MathCAD are put into evidence: simplicity of program, decrease of experimental errors, and prediction of theoretical dependencies for the determined physical parameters.
Keywords: interpolation, MathCAD, processing, experimental data, physical laws.

1. Introduction

Experimental data obtained from measurements of different physical parameters are affected by errors, depending on the quality of measuring instruments and devices, experimenter’s experience, external factors, etc [1-4]. Most often, experimental errors are randomly distributed, giving values placed nearly the theoretical curve describing the dependence between the determined parameters. Interpolation is the operation of tracing an experimental dependence curve, between experimental points, after the rule given by majority of points [5, 6]. It is usually made by the experimenter, visually, its accuracy depending on the experimenter’s experience. It can be done more rigorously with MathCAD, software containing special interpolation functions/commands – taking into account the characteristics of randomly distributed errors [7-11]. The work presents these functions, by simulating a real situation of obtaining experimental data with errors less than 10%, obtained in usual measurements in school/university labs, by inexperienced experimenters.

2. The MathCAD Program

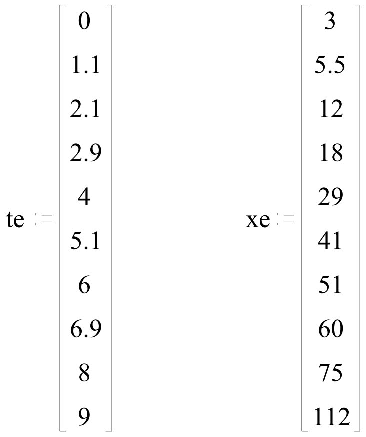
We will simulate the obtaining of some experimental data regarding a quadratic dependence of a mobile's coordinate, as a function of moving time. Fist, a time row is defined, having values from 0 to 9 (seconds), with step 1 (second). These values can be considered those determined with perfect accuracy (0% relative errors). The character „:=" is obtained by typing „:"; the character „ .." is obtained by typing „;".

$$t := 0,1\ ..\ 9 \tag{1}$$

Then, a function *x(t)* is defined, giving the values (in meters) determined with perfect accuracy (0% relative errors).

$$x(t) := t^2 + 2t + 3 \tag{2}$$

We will define two vectors, *te* and *xe*, containing experimental data with errors distributed within 10% (figure 1).



$$te := \begin{bmatrix} 0 \\ 1.1 \\ 2.1 \\ 2.9 \\ 4 \\ 5.1 \\ 6 \\ 6.9 \\ 8 \\ 9 \end{bmatrix} \qquad xe := \begin{bmatrix} 3 \\ 5.5 \\ 12 \\ 18 \\ 29 \\ 41 \\ 51 \\ 60 \\ 75 \\ 112 \end{bmatrix}$$

FIGURE 1. Input experimental data.

Generation of functions values in Figure 1 was obtained by typing "=". Generation of vectors in figure 1 was obtained by typing "ALT+m" and introducing in the dialog box the number of columns 1, and number of rows 10, respectively.

Now fallows the interpolation program.

$$xs1 := \text{cspline}(te,xe) \tag{3}$$

$$xs2 := \text{pspline}(te,xe) \tag{4}$$

$$xs3 := \text{lspline}(te,xe) \tag{5}$$

Three variables are defined and characterized by the values of three functions: cspline, pspline and lspline. Each function returns a vector of second derivatives for data vectors *te* and *xe*. This vector becomes the first argument of the interpolation function. The resultant spline curve is cubic, parabolic or linear at the endpoints, depending on the function.

m := slope(te,xe) (6)

b := intercept(te,xe) (7)

These functions return the slope and intercept of the line that best fits our data in a least square sense.

xregress := regress(te,xe,2) (8)

Returns a vector which interpolation uses to find the k^{th} order polynomial that best fits the *xe* and te data values.

xloess := loess (te,xe,1) (9)

Returns a vector which interpolation uses to find a set of second order polynomials that best fit a neighborhood of the *xe* and te data values. 1 is the size of the neighborhood.

I will chose a second degree function for interpolation, then define a function of parameters u and z, containing as elements a second degree function together with her first, second and third derivates.

$$F(z,u) := \begin{bmatrix} u_0 + u_1 \cdot z + u_2 \cdot z^2 \\ 1 \\ z \\ z^2 \end{bmatrix} \qquad (10)$$

I will define a vector with zero initial components.

$$vg := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (11)$$

The next function is useful when we have a set of measured xe and te values and we want to fit an arbitrary function to our data.

P := genfit(te,xe,vg,F) (12)

$$P = \begin{bmatrix} 4.173 \\ 1.046 \\ 1.116 \end{bmatrix} \qquad (13)$$

We choose a function g(r) to indicate the function best fitting the experimental data (index obtined by typing “[“).

$g(r) := F(r,P)_0$ (14)

We give values to index i (starting with 0 in MathCAD) and parameter r (that should be near the limits of t).

$$i := 0,1 \,..\, 9 \qquad r := 1,1.1 \,..\, 10 \tag{15}$$

The graphics of functions are presented in Figure 2. One can see the experimental values with squares, the ideal dependence with interrupted line and the interpolation function with line, fitting/superposing very well.

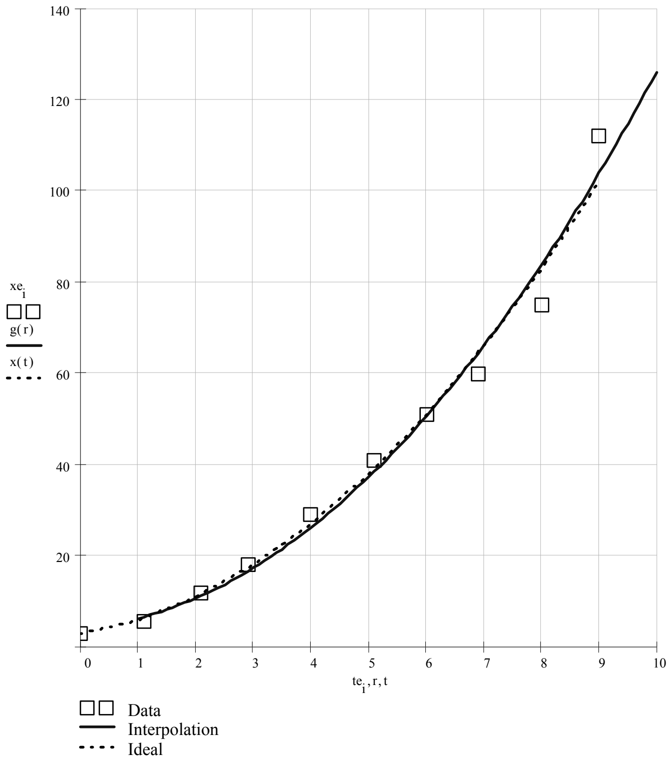


FIGURE 2. Graphics of functions.

Now we will analyze the same experimental data, but choosing a fourth degree interpolation function. The program will be the same presented before, except the command line in equation 10 (containing a fourth degree polynom) and vector in equation 11 (having 5 elements). The obtained graphics are those in Figure 3.

One can see that the two lines are not superposing, interpolation being done too close to the experimental points. This could be good for much smaller experimental errors. Therefore, the experimentator is the one that should appreciate how big the experimental errors are and which type of function could better approximate experimental data.

Comming back to the values of function P, from relation (13), best approximating experimental data, let us remark that interpolation allows the obtaining of the law describing/governing the studied physical phenomenon. Hoever, the values of the second degree polynomial parameters (1.116, 1.046, 4.173) are far away from those describing the ideal dependence (1, 2, 3). A better interpolation/law can be obtained by increasing the measurements accuracy or, more accessible, increasing number of experimental data [12].

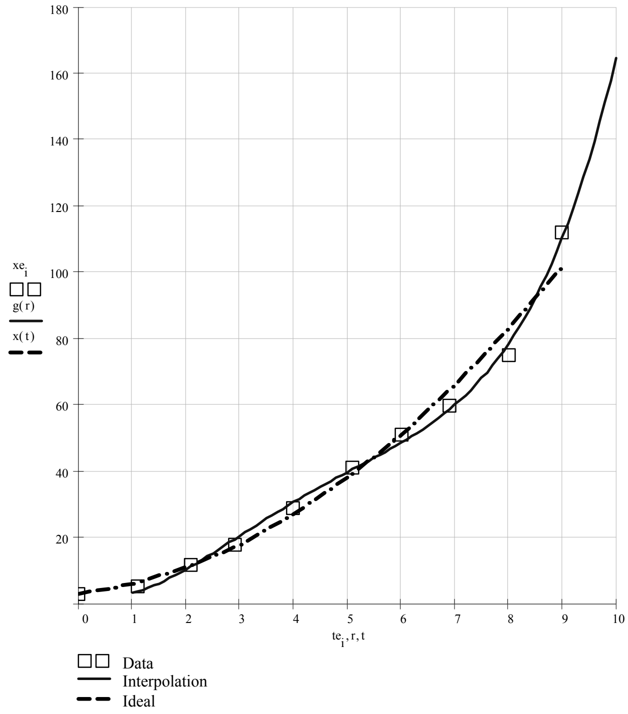


FIGURE 3. Data interpolation using a fourth degree function.

Figure 4 shows the results obtained for 50 pairs of experimental data. The obtained values of the second degree polynomial parameters (1.048, 1.781, 2.986) are very close to those describing the ideal dependence, even if the relative experimental errors are still 10%.

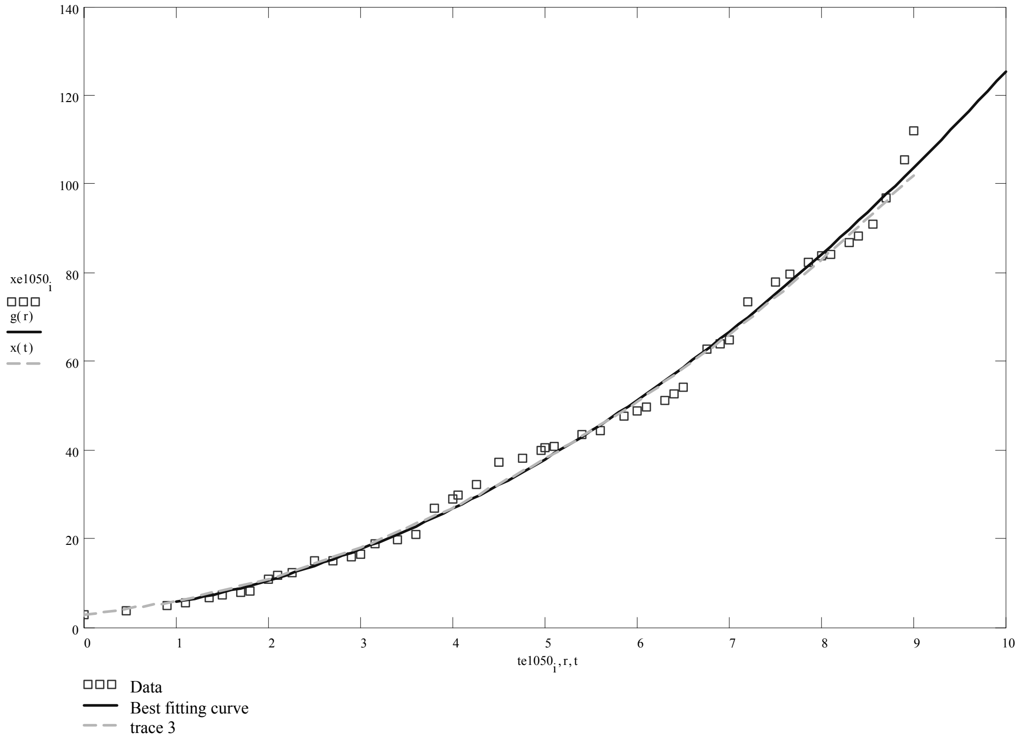


FIGURE 4. Graphics of functions for 50 pairs of experimental data.

Conclusions

MathCAD is a useful tool for interpolation of experimental data affected by relatively high errors. By interpolation are eliminated both measuring instruments and experimenter caused errors. A more detailed study shows that interpolation can lead to a strong decrease of experimental errors (for example, from 10% to 1%) and to a rigorously prediction of the governing law, by a simple increasing of experimental data number.

References

1. C. D. Buioca, *Processing of experimental data in physics*, Edyro Press, Bucureşti (2004).
2. L. Kirkup, B. Frenkel, *An introduction to uncertainty measurement,* Cambridge (2006).
3. S. G. Chapra, *Numerical methods for engineers: with software and programming applications,* McGraw-Hill Higher Education NY (2002).
4. J. D. Hoffman, *Numerical methods for eng. and scientists,* McGraw-Hill, NY (2001).
5. J. J. Risler, *Mathematical methods for CAD,* Cambridge University Press (1992).
6. L.V. Fausett, *Numerical methods using MathCAD,* Prentice Hall (2001).
7. C. D. Buioca, Proc. Nat. IT Conf., Ibra, Oman (2006), 72-77.
8. C. D. Buioca, Proc. ELSE2009, Bucharest (2009), 289-296.
9. O. Dogaru, F. Fortiş, P. Pau, H. Popa, *MathCAD*, Mitron, Timişoara (1997).
10. R.W. Larsen, *Introduction to MathCAD,* Prentice Hall (1999).
11. D. Donnelly, *MathCAD for introductory physics*, Addison Wesley (1992).
12. C. D. Buioca, L. M. Sorop, Proc. ICPS2009, Split, Croatia (2009) 82.