Online Electronic Supplement to

A Simple, All-Purpose Nonlinear Algorithm for Univariate Calibration

by

Joel Tellinghuisen
Department of Chemistry
Vanderbilt University
Nashville, Tennessee, U. S. A. 37235
tellinjb@ctrvax.vanderbilt.edu

I have chosen to illustrate the use of this algorithm with the commercially available program KaleidaGraph™ (Synergy Software, Reading, PA, USA). This program is one of several microcomputer data presentation and analysis programs that have been available for about a decade and that permit the fitting of data to user-defined functions by means of the Levenberg-Marquardt algorithm or some similar procedure.[1,2] (A few others are Axum, Igor, Origin, and SigmaPlot.) Compared with many others, KaleidaGraph is less expensive; and it is available in virtually identical forms for the Macintosh and PC platforms, with files created on one usable by the other. Its use as a data analysis tool in the physical chemistry teaching laboratory is described elsewhere (available on request).[3] For more background on the mathematical and statistical aspects of nonlinear least squares, the reader is directed to a recently published study of bias and inconsistency in nonlinear fitting, and the works cited therein.[4]

In the first example, eight calibration points were generated at integer x values 1-8 using the function $y = 1 + 5\,x + 0.01\,x^2 - 0.025\,x^3$, and were treated for constant error ($\sigma = 2.5$) and for proportional error ($\sigma = 0.14\,y$). The KaleidaGraph (KG) data sheet for these calculations is illustrated here in Fig. 1. After the x values have been entered in Column 0, the y values can be generated using a "Formula Entry" window, which is opened from the "Windows" menu. In "Formula Entry" computations, KG identifies variables by their column number, so this calculation is performed with the following entry in the window:

```
C1 = 1 + 5.*C0 + .01*C0^2 - .025*C0^3 .
```  (1)

Columns 2 and 3 contain the σ values for constant error (2.5) and proportional error, respectively; the latter are calculated using the following "Formula Entry":

```
c3=c1*.14 .
```  (2)

(Note that the column labels are not case-sensitive.)

For the computation of the error bands illustrated in Fig. 1 of the paper, the "unknown" in Row 9 has been masked out. The data are plotted by choosing "Scatter" plot under the "Gallery" menu.

Calib. Scheme data (4/5/00)

	0 x	1 y	2 sig1	3 sig2
0	1.00	5.9850	2.5000	0.83790
1	2.00	10.840	2.5000	1.5176
2	3.00	15.415	2.5000	2.1581
3	4.00	19.560	2.5000	2.7384
4	5.00	23.125	2.5000	3.2375
5	6.00	25.960	2.5000	3.6344
6	7.00	27.915	2.5000	3.9081
7	8.00	28.840	2.5000	4.0376
8				
9	9.00	20.000	2.5000	2.8000
10				

Fig. 1. KaleidaGraph data sheet for calibration examples illustrated in Figs. 1 and 2 of the published paper.

Error bars can then be added by selecting under the "Plot" menu. The curve fit is initiated by picking "General" under the "Curve Fit" menu and then selecting one of the named fits (or adding one, if necessary). By clicking on "Define" in the instruction box that opens, the user can either modify the existing fit function or enter a new one. The "Define" box also permits one to specify the convergence criterion and to select several other options, including "Weight Data." When this option is selected, the program prompts the user for the column containing the σ values for the data. In the present case, the two different weighting choices are treated by simply selecting the appropriate column ("sig1" or "sig2") in turn. The program then properly calculates the weights as $w_i = \sigma_i^{-2}$.

Figure 2 illustrates the results from three different fits of these data. The box at upper right gives parameters and their errors for a fit to the same function used to generate the "data." To obtain these results, the user need only type the following in the "Define Fit" box:

```
a + b*x + c*x^ 2 + d*x^ 3; a=1; b=1; c=1; d=1 .
```
(3)

Note that initial values must be given for each adjustable parameter; however, in the present case these values need only be nonzero, because the fit is linear (though not straight-line), so convergence is assured. The results box at lower right in the figure shows the effect of recentering the fit at $x = 8$, obtained by using

```
a + b*(x-8) + c*(x-8)^ 2 + d*(x-8)^ 3
```
(4)

as the fit function. [If this is run following the fit to Eq. (3), the initial values need not be given, as the program will still have the previous values in its memory.] Note that now the value of the calibration function at $x = 8$ is a, and the error in a is the error in the calibration function at this x. By altering the value of the offset from 8 to other values of interest, one can trace out the error band as a function of x. [The fit to Eq. (3) has already given the error at $x = 0$, again as the error in a.]

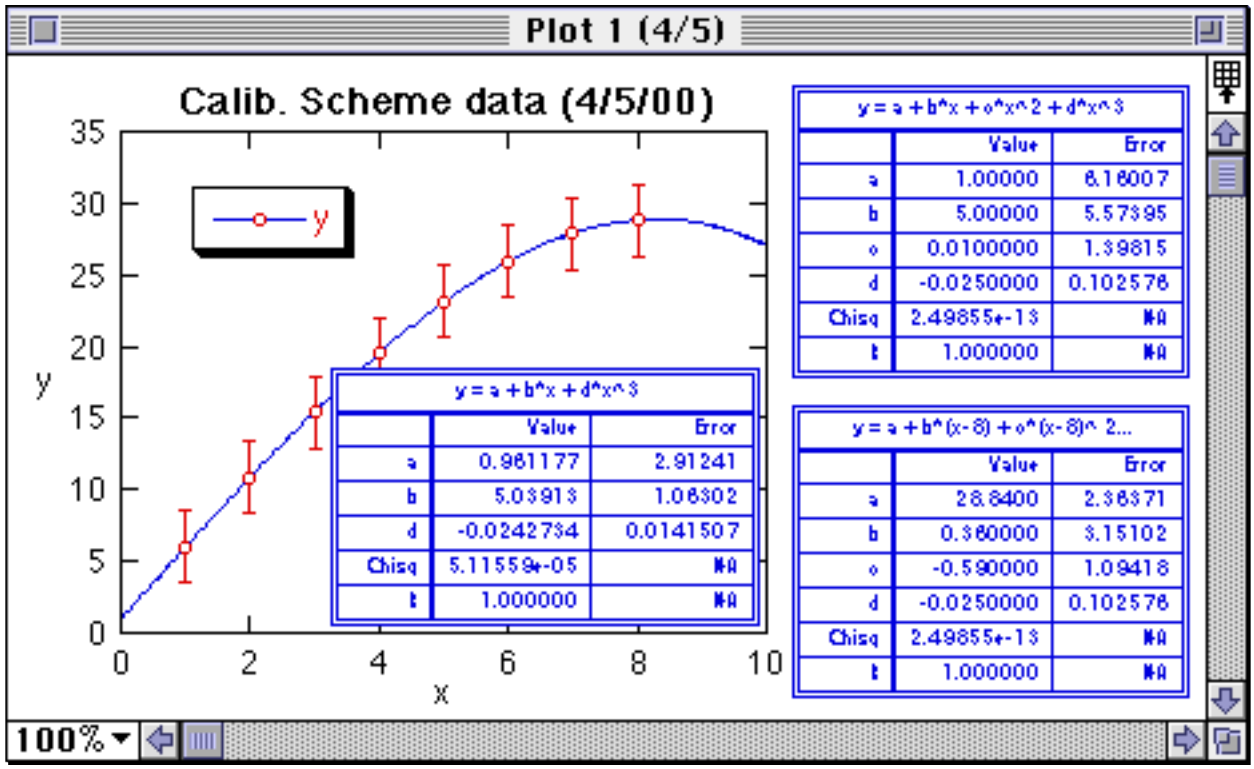

Fig. 2. Calibration plot shown for constant error, including results from three different user-defined least-squares fits.

The third results box (inside the figure axes) shows the effect of dropping the quadratic term from Eq. (3). This term was intentionally made statistically insignificant for the sake of this illustration. After the deletion of the quadratic term, the intercept still remains uncertain (1 σ) by three times its magnitude and might also be dropped from the fit model. Whether this is appropriate or not depends on the situation. If the fit model is based on some accepted theory, in which the parameters have physical significance, such deletions can lead to systematic errors in the remaining parameters and their errors. Hence, if the parameters and their errors are a goal of the analysis, such deletions should be avoided, even if, as here, the values are statistically insignificant. On the other hand, much calibration fitting is *ad hoc* in nature, with terms included to improve the quality and reliability of the fit. The goal of the fit is the fit function itself, not its parameters. In such a case, the deletion of insignificant parameters is warranted. For example, in the calibration of fluorescence data, an intercept might be included to allow for interferences from unknown contaminants in the samples and calibrants. The observation of a statistically undefined intercept can be taken as *prima facie* evidence that such interferences are not a problem, and dropping the intercept from the fit is then justified.

The quantity "Chisq" in the fit results boxes is the chi-square value, defined as $\chi^2 = \Sigma\, w_i\, \delta_i^2 = \Sigma\, (F_i/\sigma_i)^2$, where δ_i is the residual for the ith point, given in the present case by F_i from Eq. (8) in the paper. Since these fitted "data" are exact, the value of χ^2 is of no interest here (except perhaps to

show the slight mismatch when the quadratic term is dropped). However, for actual data, its behavior would provide additional guidance on the matter of keeping or dropping terms from the fit model. The statistical properties of χ^2 in least-squares fitting are discussed elsewhere.[1,2,4] For present purposes it suffices to note that rounding or otherwise altering a fit parameter by $^1/_3$ of its standard error produces an insignificant increase in χ^2.[5] Thus, dropping the intercept in the case just discussed is supported also by statistical considerations of χ^2.

Next consider estimating the unknown concentration x_0 and its error. For this purpose I will use the exponential function defined in Eq. (11) of the paper, for the case of proportional error in y ("sig2"). Row 9 in the data sheet must first be unmasked for inclusion in the plot and fit. The dummy x value in this row just serves to differentiate between this value and the calibration points, as is noted in the paper. Since the fit is now a truly nonlinear one, the user must take care in the choice of initial parameter values, or the fit may diverge.

The results of such a fit (Fig. 3) yield a value of 4.19(91) for x_0 and its standard error. As before, the intercept a in this fit is statistically negligible. Dropping it from the model leads to the results in Fig. 4, from which x_0 = 4.20(87). The two models thus differ negligibly in their estimation of the unknown in this case. In fact, this relationship remains true except in the extrapolation region at low x, where the latter model (no a) yields much more precise estimates, as expected.

As has been noted in the paper, multiple unknowns can be accommodated easily by just entering in the data sheet additional rows like row 9, each containing a dummy x value and a measured y (y_0) and its error. Each such row is unmasked in turn, and the fit is rerun by clicking in an "update" box on the data sheet.

y = (x>8)?(a+b*(1-exp(-c*f))...

	Value	Error
a	-0.148968	2.34575
b	41.8234	15.2043
c	0.157038	0.108784
f	4.18577	0.907113
Chisq	0.156492	NA
R	0.999756	NA

Fig. 3. Results from a fit of the calibration data and "unknown" to Eq. (11) in the paper. The unknown is f.

y = (x>8)?(b*(1-exp(-c*f))):...

	Value	Error
b	42.5464	11.1296
c	0.151139	0.0548614
f	4.20157	0.871858
Chisq	0.160719	NA
R	0.999750	NA

Fig. 4. Rerun of fit of Fig. 3 without intercept parameter.

[The user new to KG should note that the default labels for adjustable fit parameters are $m1$–$m9$. If the Macro Library is loaded during installation, the alternate definitions $a = m1$, $b = m2$, $c = m3$, and $d = m4$ will be in place; but the user will need to enter in the library the definitions for parameters beyond $m4$, *e.g.* the use of f (= $m5$) in the fits that yielded Figs. 3 and 4.]

y = a + b*x		
	Value	Error
a	12.6429	0.324554
b	0.398143	0.00535984
Chisq	9.65257	NA
R	0.998553	NA

Fig. 5. Results from unweighted fit of "SC" data in Table 1 of Ref. 6.

In all of the calculations discussed up to this point, the y-error in the data has been assumed to be known *a priori*. The final example in the paper treats actual data and assumes that the y-error is assessed *a posteriori*, from the data themselves. The data of note are the 18 standard calibration points from Table 1 of the paper by Campaña, *et al.*,[6] describing their program ALAMIN. The "General" routine in KG is again used to carry out an unweighted fit of these data to a straight line. (This fit can be done by selecting "Linear" under the Curve Fit menu; however, only the "General" option provides error estimates for the parameters.) The results (Fig. 5) show parameters and errors in agreement with those shown in Fig. 1 of Ref. 6. In an unweighted fit, KG assumes $\sigma = 1$ in computing "Chisq." Thus, this quantity becomes just $\Sigma\, \delta_i^2$, from which the variance in y is estimated in the usual way, as $s_y^2 = (\Sigma\, \delta_i^2)/\nu$, where ν (degrees of freedom) = $n - p$ = 18 (# points) – 2 (# parameters) here. The result for s_y is 0.77671, in agreement with the quantity S_s in Fig. 1 of Ref. 6.

y = a + b*(x-40)		
	Value	Error
a	28.5686	0.190758
b	0.398143	0.00535984
Chisq	9.65257	NA
R	0.998553	NA

Fig. 6. Same fit as in Fig. 5, recentered at x = 40.

Other quantities in this figure of Ref. 6 can be reproduced by using the polynomial recentering method and the new algorithm. For example, "Rp" and "S(R)" are obtained as a and $(s_a^2 + s_y^2)^{1/2}$, respectively, using the recentering method, *i.e.*, fitting to $y = a + b\,(x - 20)$, $y = a + b\,(x - 40)$, *etc.* Sample results are shown in Fig. 6. The quantities "Sc" are the estimated errors in x_0 for y_0 values obtained by averaging each group of three calibrants at a given c in Table 1 of Ref. 6. These can be obtained using the new algorithm by running either (1) a weighted fit with all 18 calibration values given σ values = s_y and each y_0 value in turn given $\sigma = s_y/\sqrt{3}$, or (2) an unweighted fit having each y_0 entered in triplicate. In the latter case, however, KG miscounts the degrees of freedom (18 instead of 16), so the resulting errors in x_0 must be scaled by $\sqrt{9/8}$ to correct for this. The reason for this need for different treatments for weighted and unweighted fits is that for the former, KG assumes *a priori* data errors and computes $\mathbf{V} = \mathbf{A}^{-1}$, as given in Eq. (3) in the paper, while for unweighted fits, it assumes *a posteriori* errors, takes weights = 1, and

y = (x>100)?(a+b*c):(a+b*x)		
	Value	Error
a	12.6429	0.324552
b	0.398143	0.00535981
c	59.4187	1.22314
Chisq	16.0002	NA
R	0.998570	NA

Fig. 7. Determination of "Sc" (= 1.2231) for "c3" in Ref. 6.

computes $\mathbf{V} = s_y^2\,\mathbf{A}^{-1}$, with s_y^2 evaluated as already noted, as "Chisq"/ν. Some typical results for the "weighted" approach are illustrated in Fig. 7.

References

1. P. R. Bevington, *Data Reduction and Error Analysis for the Physical Sciences* 1969, McGraw-Hill, New York; Chpt. 11.
2. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, Cambridge Univ. Press, Cambridge, U. K., 1986; Chpt 14.
3. J. Tellinghuisen, *J. Chem. Educ.*, 2000 (in press); with online supplement.
4. J. Tellinghuisen, *J. Phys. Chem. A*, 2000, **104**, 2834.
5. J. Tellinghuisen, *J. Mol. Spectrosc.*, 1995, **173**, 308.
6. A. M. G. Campaña, L. C. Rodríguez, F. A. Barrero, M. R. Ceba, and J. L. S. Fernández, *TRAC*, 1997, **16**, 381.