Is there still a place for linearization in the chemistry curriculum?

Andrew R. McCluskey*

European Spallation Source ERIC, Ole Maaløes vej 3, 2200 København N, DK

E-mail: andrew.mccluskey@ess.eu

Abstract

The use of mathematical transformations to reduce non-linear functions to linear problems, which can be tackled with analytical linear regression, is commonplace in the chemistry curriculum. The linearization procedure, however, assumes an incorrect statistical model for real experimental data; leading to biased estimates of regression parameters. As, non-linear optimization is more accessible than ever with modern computing by introducing linearization, without a detailed discussion of the shortcomings, we are failing to equip students with the correct tools for formal data analysis. I hope that this commentary will start a discussion in the community around the place of linearization in the chemistry curriculum.

In chemistry, non-linear relationships are commonly found between dependent and independent variables. These relationships can be simplified by the process of "linearization", where some mathematical transformation is used to reduce the non-linear problem to a linear one. By linearizing a function, analytical linear regression can be used to quantify parameters of interest, rather than relying on numerical optimisation. We see this process in chemistry textbooks[1,2] and undergraduate degree programs: for example where it is applied to first- and second-order rate equations, and the Clausius-Clapeyron and Arrhenius equations.[1,3,4]

While mathematically sound for noise-free measurements, linearization can introduce errors in the analysis process for real experimental data. Specifically, it can lead to biased estimates of regression parameters; the gradient and intercept of the straight line. Therefore in formal analysis, the use of linearization should be avoided. However, because linearization is included in a general chemistry education, without discussion of its problems, it is regularly found in research publications. Although not analytically tractable, non-linear optimisation is now accessible through standard analysis software and programming languages and lacks the problems of linearization.

To exemplify the problem that results from linearization, we can consider the decomposition of hydrogen peroxide H_2O_2 in the presence of excess cerium(III) ion, which follows first-order rate kinetics with the form[1]

$$[H_2O_2]_t = [H_2O_2]_0 \exp(-kt), \tag{1}$$

where, $[H_2O_2]_t$ is the concentration of hydrogen peroxide at time t, $[H_2O_2]_0$ is the initial concentration and k is the rate constant (representative data is shown in Fig. 1a). Linearization of Eqn. 1 involves taking the natural logarithm of both sides to produce

$$\ln [H_2O_2]_t = -kt + \ln [H_2O_2]_0. \tag{2}$$

The gradient and intercept from linear regression, of $\ln [H_2O_2]_t$ on t, are therefore equal to $-k$ and $\ln [H_2O_2]_0$, respectively (Fig. 1b).

If we were to perform repeated measurements of the concentration of H_2O_2 as a function of reaction time and analyse each repeat, we can build up a distribution of estimates of k (Fig. 1c & 1d). The simplest way to analyse a linearized function is by ordinary least squares (OLS) linear regression, which we can compare with unweighted non-linear optimization. Non-linear fitting gives a normal distribution of estimated values of k, with a mean centred on the true value, i.e. the estimation is unbiased. The linearized form, however, gives a

biased, broad, asymmetrical distribution, where the normalised mean is 1.05 . The linearized approach will, on average, overestimate the value of k and any single estimate of k has a higher chance of being further from the true value.

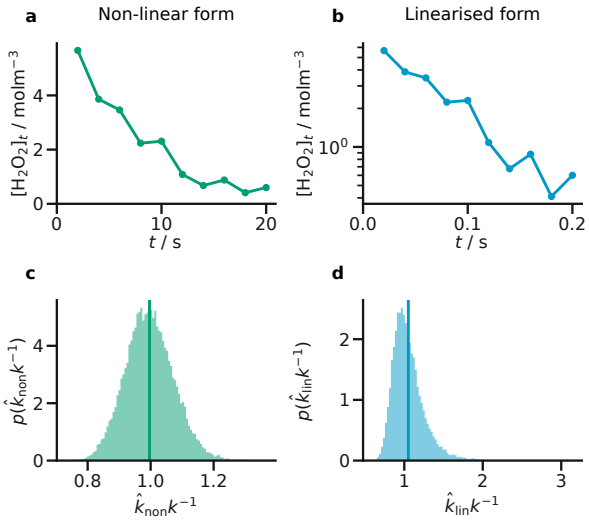

Figure 1: Representative data for first-order integrated rate equation, with a true value of $k = 0.15\,\mathrm{s}^{-1}$ and $[\mathrm{A}]_0 = 7.5\,\mathrm{molm}^{-3}$, showing (a) the non-linear and (b) the linearized forms. Estimates of k, normalised to the true value of k, from 2^{15} analyses of unique representative datasets, using (c) unweighted non-linear fitting and (d) linearization followed by ordinary least squares, with the vertical lines indicating the distribution means.

By using OLS or unweighted non-linear optimisation, we are assuming that the uncertainties in our data are all the same, i.e., they are homoscedastic. It was noted by Perrin,[3] however, these homoscedastic uncertainties may become heteroscedastic as a result of the linearization process (note the error bars in Fig. 2b). Therefore, the use of OLS for linearized data is insufficient, instead weighted least squares (WLS), where the weights are the correctly propagated measured uncertainty, should be used. For the example in Eqn. 2, the correct error propagation is to divide the measured error by the nominal value (Fig. 2b). WLS leads to a normal distribution of estimated k, but still, the distribution is biased (Fig. 2d), with a normalised mean of 0.95 . Non-linear optimisation, meanwhile, still produces an unbiased estimate. Even when the errors are correctly propagated and included in the analysis, the linearization approach will give a biased estimate of k.

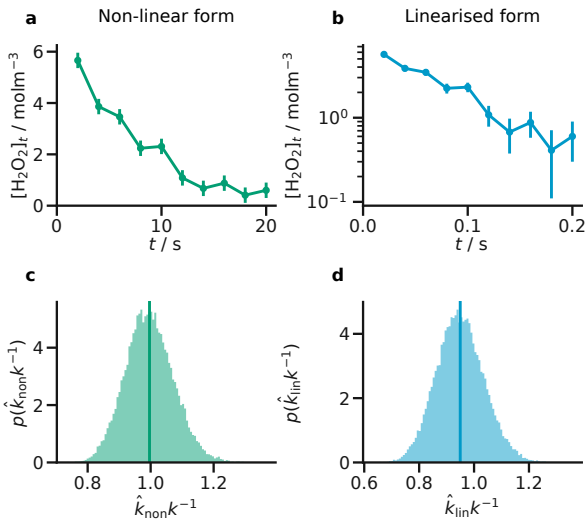

Figure 2: The same representative data as Fig. 1 with error bars of 0.3 molm^{-3} (a & b). The same number of analyses were performed, however, this time using (c) weighted non-linear optimisation and (d) weighted least squares with propagated uncertainties, the vertical lines indicate the mean of the distribution.

The observed bias can be understood by recognising that the measurement of any variable, y, is only ever an estimate of the true value, $\hat{y}$, which is a random draw from a distribution of values, $p(y)$. The shape of this distribution depends on the noise or uncertainty in the measurement. It is commonly assumed that random uncertainty sources will lead to a normal distribution, $p(y) \sim \mathcal{N}(\mu, \sigma^2)$, which is defined by the mean, μ, and standard deviation, σ (Fig. 3).[1] When linearization is used, a mathematical transformation is performed on the dependent variable and if that transformation scales in a non-linear fashion, i.e., the reciprocal or logarithm is taken, it will cause the normally distributed variable to become non-normal (Figs. 3b & 3c).

For normally distributed variables, both OLS and WLS produce unbiased estimates of the regression parameters. However, this is not the case when non-normally distributed variables are used. By applying OLS or WLS to non-normally distributed variables, we use the wrong statistical model for our analysis. However, the correct statistical model is being applied in the non-linear optimisation case, where the variables have not been transformed and are therefore still normally distributed.

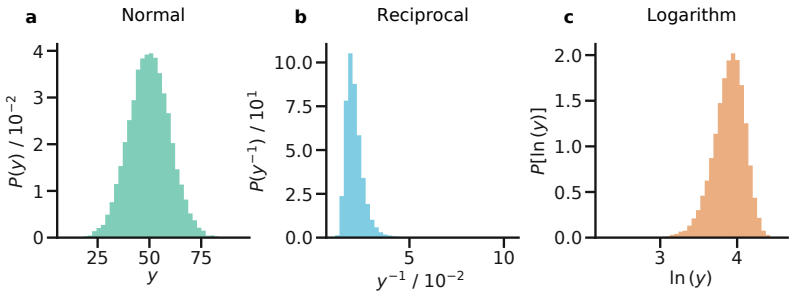

Figure 3: Histograms showing the effect on (a) a normal distribution, of mathematical transformations that scale non-linearly: (b) the reciprocal or (c) the logarithm. Produced from 2^{15} random samples from the normal distribution $\mathcal{N}(50, 10^2)$.

The linearization process has no place in formal analysis given its failure to produce unbiased parameter estimates. Yet, due in part to the complexity of the more robust non-linear optimisation, linearization is still taught regularly to chemistry students. As, in a classroom or exam hall, it is feasible for a student, equipped with graph paper and a ruler to estimate the gradient and intercept of a straight line or perform some qualitative analysis. However, by failing to show the problems with linearization, we fail to adequately provide students with the skills to evaluate the robustness of chemical data analysis.

Recent developments in computing and access to programming and tools, such as the Jupyter Notebook[5] or the Solver functionality in Microsoft Excel, mean that non-linear optimisation is more accessible than ever. Therefore, I believe that the deficiencies of linearization should be taught alongside the non-linear optimisation solution. In addition to reducing the use of this flawed process in the chemical research literature, this will give students a more rounded understanding of data analysis, while keeping the utility of "quick" analyses with linearization.

Data availability

Electronic Supplementary Information (ESI) available: A complete set of analysis/plotting scripts allowing for a fully reproducible and automated analysis workflow, using showyour-

work,[6] for this work and a Jupyter Notebook showing the use of weighted non-linear optimisation for representative first-order rate kinetics data is available at `https://github.com/arm61/against-linearisation` (DOI: 10.5281/zenodo.xxxxxxx) under an MIT license, while the text is shared under a CC BY-SA 4.0 license.[7]

Acknowledgements

The author thanks Benjamin J. Morgan, Samuel W. Coles, Thomas Holm Rod, Gabriel Krenzer, and Kasper Tolborg for the insightful discussion that led to this work. Additionally, the author would like to thank those that engaged in discussion on Twitter, in particular Carl Poree and Fiona Dickinson, when the problem of linearization in Arrhenius modelling was initially raised.

References

(1) Monk, P.; Munro, L. J. *Maths for Chemistry: A chemist's toolkit of calculations*, 2nd ed.; Oxford University Press: London, UK, 2010.

(2) Atkins, P.; de Paula, J.; Keeler, J. *Atkins' Physical Chemistry*, 11th ed.; Oxford University Press: London, UK, 2018.

(3) Perrin, C. L. Linear or Nonlinear Least-Squares Analysis of Kinetic Data? *Journal of Chemical Education* **2017**, *94*, 669–672.

(4) Harper, J. K.; Heider, E. C. Data Linearization Activity for Undergraduate Analytical Chemistry Lectures. *Journal of Chemical Education* **2017**, *94*, 610–614.

(5) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C.

Jupyter Notebooks – a publishing format for reproducible computational workflows. 2016.

(6) Luger, R. showyourwork. https://github.com/rodluger/showyourwork, 2021.

(7) McCluskey, A. R. against-linearisation-0.0.x. https://github.com/arm61/msd-errors, 2023.