A Student's Guide to Data and Error Analysis, *H. J. C. Berendsen,* Cambridge Univ. Press, 2011, 225 p., ISBN: 978-0-521-13492-7 (pbk), $29.99.

In the past two or three decades, the subject of "Data and Error Analysis" has undergone two separate but equally important revolutions. The first is technological: The power of modern software and hardware is such that tasks that once were PhD-scale projects may now be taught to beginning undergraduates in a few minutes. The second is conceptual: Bayesian methods have revolutionized analyses of statistics-limited data (the discovery of planets around distant suns is one example). Further, they have led to a unification of and logical way of presenting scientific analysis and have, to a large extent, supplanted earlier discussions of how the "scientific method" really works.

Against this backdrop of dramatic change, the evolution of laboratory courses and exercises in the undergraduate physics program has been much more sedate, and, as one who has been involved in such instruction, I worry that our students are not being well prepared for the brave new world we now live in. In trying to improve such instruction, the need for new books is acute, and H. J. C. Berendsen's little paperback is an interesting addition. Among the book's considerable virtues are that it is short, inexpensive, and friendly---all features that students will like. But your ultimate opinion about this book will likely depend more on how you react to two decisions the author has taken to deal with advances in technology and concepts described above.

First, the author, a pioneer in the development of software for molecular dynamics simulations, has based this book on software routines written in the open-source language Python (with the NumPy and SciPy extensions). The 27 pages of listings represent about 10% of the total pages in the book. Unfortunately, the Python codes and other supplements to the book, while promised on the author's website, are currently (February 2012) not yet available. Graphics are to be handled by a separate program---the author recommends GNUplot (or one of its adaptations for Python or R) or their own simple plot package, again not yet available at the time of this review. Since Python is open source and quite powerful, any analysis an undergraduate needs is readily available, as are many state-of-the-art methods. The disadvantage is that Python and the graphics packages are based on command-line programming, with a need to write or understand low-level commands in order to produce acceptable results (e.g., to graph data). Commercial software such as Matlab, LabVIEW, or Igor are much better in providing graphical environments in which graphics and analysis may be invoked. Igor is a personal favourite because, through menu-driven choices, it can generate the kind of code used in this book, which can then be tweaked and reapplied to different analyses. It has some of the advantages of languages such as Python without the need to actually learn much programming.

Second, Bayesian methods do make an appearance, but only briefly, in the last chapter. Thus, the bulk of the presentation is traditional, with only hints of deeper justification at the end. Indeed, the author often takes a "here's the formula -- we'll explain later" approach.

While many physicists will instinctively think that this is not a good way to proceed, my own experience in teaching lab courses has shown me that there is some merit in such an approach. In particular, since higher-level courses often return to the same problems of analysis, there is the chance to first "teach the recipe" and then later to explain it. For some students, this method can be effective. However, for reasons discussed more in the online version of this review, I feel that too much is lost in putting the conceptual foundations at the end, and I prefer the approach taken, for example, by D. S. Sivia in his book "*Data Analysis: A Bayesian Tutorial*." The latter is too advanced for an introductory class. Nevertheless, I believe Berendsen's book to be an improvement on the one we currently use in our introductory lab courses at Simon Fraser University (J. R. Taylor, *An Introduction to Error Analysis*), which, while well written, is badly out of date.

One topic that Berendsen does introduce---but I wish he did more!---is Monte Carlo analysis. In *Numerical Recipes* by Press *et al*., a book filled with much useful discussion of data analysis, the authors make the claim that "Offered the choice between mastery of a five-foot shelf of analytical statistics books and middling ability at performing statistical Monte Carlo simulations, we would surely choose to have the latter skill". I believe that this statement applies to undergraduate instruction, as well, with the implication that we should be teaching Monte Carlo simulation all throughout the undergraduate program. At the lower levels, such simulations can be as simple as drawing samples from a given distribution and histogramming or otherwise analyzing them. The sophistication of programs can then grow in step with the sophistication of analyses attempted. As a rule of thumb, students should be able to generate simulated data for any analysis that they attempt. If we teach nothing more than the idea that it is crazy to analyze real data without first testing on simulated data, we will have made a significant advance in preparing our students for the "real" world. Such thoughts are implicit in the discussions and example codes given in the book. I liked, for example, the discussion of bootstrap methods. I just wish that there were more such examples and that their motivation were stated more directly and forcefully.

Some sections seem rather dated. For example, there is a Chapter on graphical analysis of data with errors. On the one hand, it is certainly true that it is worth exploring data qualitatively by hand before doing a more sophisticated analysis and that computer programs (especially the commercial ones that I mention above) can do this interactively in an intuitive way. On the other hand, the emphasis on trying to figure out transformations that lead to linear plots is a dangerous game. Since errors transform nonlinearly and since it is hard to do hand fits that take into account such weighting, the result of a hand fit can be misleading. Also, students can waste time worrying about issues that are artifacts of such transformations. As an example, consider a fit to an exponential with background, $y = a \exp(-x) + b$. To plot this on a straight line involves estimating b, subtracting it from the data, and then plotting log (y-b) vs x. Students invariably (and rightly) worry about the points they "lose" when they take the log of points that are below the background level b. Of course, in a proper approach (*e.g*., a nonlinear least-squares fit to the original function), such a problem does not arise.

Another part that seemed out of date is the appendix filled with long tables of numbers for distributions and the like. The effort gone into preparing such tables in the "Scientific data"

section would have been far better spent in making up some interactive code that would illustrate the different distributions on computer. Such illustrations are rather trivial to write in almost any modern language (see the "High School Statistics" section of the Wolfram Demonstrations website for a number of these, for example).

The curve-fitting chapter is one that also seems out of date in places. First, one should recognize that advances in curve-fitting software can simplify significantly the types of material that need to be presented. Traditional discussions of curve fitting, including this one, distinguish between linear and nonlinear regression. The traditional justification was two-fold. First, if the fit function is linear in the unknown parameters, the best values of the parameters may be determined directly using linear algebra. By contrast, nonlinear fits depend on iterative functions that often require the user to choose starting values. A second justification is that one can prove theorems about goodness of fit and sampling distributions for linear fits. For nonlinear fits, the theorems are usually approximately correct, but there are some notable exceptions.

To assess these reasons in the present day, we should recognize that there is little reason to teach the linear-algebra methods for solving linear fits, particularly in an introductory course. On a practical level, one can use a nonlinear solver in all cases. While they are less efficient numerically, the differences are likely to be unnoticeable. Moreover, for linear fits, the chi-square surface is convex, meaning that iterative methods will converge from arbitrary starting points. Thus, one can substitute the somewhat heavy dose of linear algebra involved in the traditional method with a pictorial presentation of the gradient methods used to minimize the chi square statistic in nonlinear fits.

John Bechhoefer
Simon Fraser University