1 / VOL. 1, NO. 1
THE CHEMICAL EDUCATOR

ISSN 1430-4171
http://journals.springer-ny.com/chedr
10.1007/s00897960012a

Computers in Chemistry

Spreadsheet Approach to the Linear Least Squares Fit

MARIE L. CARMAN and **THOMAS G. CHASTEEN**
Sam Houston State University
Huntsville, TX 77341-2117, USA
chm_tgc@shsu.edu

The hands-on nature of this experiment is best served by having students program a simple spreadsheet themselves.

A method of programming two commercially available spreadsheet programs to calculate the linear-least-squares (LLS) line from a set of sample data is presented. Student data from a Quantitative Analysis chemistry course are used as examples. Files for both Excel for Macintosh and Quattro Pro for IBM compatibles are available online with this paper. These include blank spreadsheet templates, an example using seven *xy* pairs of student data with graphs of the data and best-fit line, and more advanced spreadsheet template files that incorporate the calculated standard deviation in the regression line's slope and *y*-intercept, as well as the error about the line.

The determination of the linear least squares (LLS) fit of, for instance, experimental calibration data is routinely used in analytical chemistry courses on the sophomore and junior levels. The actual calculation of the best-fit line can be done by hand; however, if the use of many calibration points or a display of the

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

line along with the calibration data is required, the time spent calculating and plotting the least squares line may be better spent, for instance, writing the report or determining unknown concentrations. The recent advent of graphing calculators has meant that many students have access to an almost instantaneous LLS function that will handle many points and give the sample correlation coefficient. The *complete* removal of the student from the workings of the LLS calculations, however, is a tradeoff that many instructors are loath to make. Moreover, getting a printout from graphing calculators, though possible, is not trivial. The method described here is an effort to keep the students' hands in the LLS calculation, thereby helping them to better understand its workings. It still allows them to quickly process relatively large data sets, calculate the quality of the line fit (i.e., determine the correlation coefficient, r), and finally to create a spreadsheet template for use later. In our Quantitative Analysis class, these calculations are common for chromatographic calibrations (the example given here) and Beer's Law plots.

Computer Programs

The hands-on nature of this experiment is best served by having students program a simple spreadsheet themselves in order to perform the classical LLS calculation that determines the best-fit line and plots a graph of the LLS fit. To this end, common spreadsheet programs such as Microsoft Excel® for IBM compatibles or Macintosh, Borland's Quattro Pro® for IBM compatibles or Lotus Development's 1-2-3 for IBM compatibles are all suited to allow even the computer uninitiated student to program the calculation and plot the results. Though almost any spreadsheet program can be programmed to perform this calculation, the plotting of the results is actually not handled very well by some (as compared to dedicated graphing programs, e.g., Synergy Software's Kaleidagraph® for the Macintosh). Since we feel that the output of the LLS plot is an important part of the learning process, we believe that care must be exercised in selecting a spreadsheet that can relatively easily yield that line along with the data from which it was calculated. In this way, a student completes the process of data input, calculation, and graphical output. Seeing the results is an important part of the learning process.

Finally, just as a complete "black box" calculation can be performed with a pre-programmed graphing calculator, most spreadsheets have LLS functions of their own. Though these functions could be used as a check—for slope, y-intercept, and correlation coefficient—avoiding the programming and statistical manipulation by solely using the "hardwired"

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

functions defeats the purpose of this lesson and we have avoided suggesting this in our Quantitative Analysis laboratory.

We have included spreadsheet examples for both Quattro Pro for IBM compatibles (version 5.0) and Excel for the Macintosh (version 4.0). The variations between common equations used in different spreadsheet programs have (finally) become relatively minor and careful use of online help and program manuals will usually solve problems arising from the differences in equation formats among programs (or even among different versions of the same program). We have made some effort here to describe important differences in programming for the two programs we describe; however, a more generalized description of common spreadsheet programming differences is available [1].

Statistical Background

The LLS calculation (regression analysis) described here generates an equation for a line that is determined by minimizing the absolute differences (in the y direction) between sample data points and the line [1], [2], [3], [4]. Inherent in the process detailed here are the assumptions that (1) the greatest variation in the data lies in the y direction, the dependent, that is, the measured variable; (2) the absolute variation in y values are comparable; and, of course, (3) the relationship between x and y is linear. More complex and exacting approaches to this process have been detailed by other workers, specifically for cases in which significant error is present in *both* x and y data [2], [5], [6], [7], when an iterative approach to determining the slope and intercept is useful [8], or when a weighted least-squares approach is warranted [2].

Generating the components of the equation for the best-fit line, $y = mx + b$, where m is the slope and b is the y intercept, involves the following calculations:

$$m(\text{slope}) = \frac{n\Sigma x_i y_i - (\Sigma x_i)(\Sigma y_i)}{n\Sigma(x_i^2) - (\Sigma x_i)^2}$$

and

$$b(\text{intercept}) = \frac{(\Sigma y_i)(\Sigma x_i^2) - (\Sigma x_i)(\Sigma x_i y_i)}{n\Sigma(x_i^2) - (\Sigma x_i)^2}$$

where n is the number of xy data pairs and x_i and y_i represent one of those pairs. Note that from a practical point of view it is easy to confuse the sum of (x_i squared) with the

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

(sum of x_i) squared, that is, $\Sigma(x_i^2)$ or $(\Sigma x_i)^2$, so this must be approached carefully. We hope that the online files available here will help alleviate confusion in this regard.

The process of determining the equation for the best-fit line also involves determining the sample correlation coefficient, r, which is based on three values derived from the xy data set: S_x, S_y, and S_{xy} which are the standard deviation in x, standard deviation in y, and the covariance of x and y, respectively. These values can be calculated as follows:

$$S_x = \sqrt{\frac{\Sigma(x_i - \bar{x})^2}{n-1}} \qquad \text{where } \bar{x} = \text{mean of the } x \text{ values}$$

$$S_y = \sqrt{\frac{\Sigma(y_i - \bar{y})^2}{n-1}} \qquad \text{where } \bar{y} = \text{mean of the } y \text{ values}$$

and

$$S_{xy} = \frac{\Sigma(x_i—x)(y_i—y)}{n-1}\,.$$

Finally, the sample correlation coefficient is rather simply calculated from the following:

$$r = \frac{S_{xy}}{S_x S_y}\,.$$

Determining the standard deviation in the slope and y-intercept and the random error about the line may also be desired. This can be easily calculated using a spreadsheet. The equations for these calculations are in Appendix A.

Intimidating as these equations may seem to the average quantitative analysis student, the actual spreadsheet programming, in the main, is less complex than it first appears. It involves determining the sum of columns of numbers (e.g., the Σx_i, Σy_i, $\Sigma x_i y_i$, and Σx_i^2 from the data pairs) and a few complex equations which can be performed in steps, that is, the results calculated in one cell are then used in another. A few helpful spreadsheet functions like COUNT (in Excel and Quattro Pro) add the ability for the number of xy data pairs to be counted automatically instead of being entered manually. When we last taught this method, in the Fall of 1995, we left the standard deviation calculations out (see Appendix A). If the laboratory is performed over a number of sessions, the more complicated standard deviation programming can come later, once students are more familiar with the spreadsheet.

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

Spreadsheet Programming

The downloadable files that have been included with this paper are of three types: linear-least-squares spreadsheet template files (template.wbl and template for Quattro Pro and Excel respectively), spreadsheet files containing the data (octane.wb1 and octane), graph files containing the plots of the student data (octgraph.wbl and octane graph). Appendix A includes the equations for the calculation of the standard deviation in the slope and *y*-intercept and the error about the line, so, finally, we have included two additional data files (eroctane.wbl and octane error) with these additional calculations incorporated into the spreadsheet. The template files (Figure 1 is a selected region from the Excel student template file) have no data in the *x* and *y* data columns and no conditional formulas (see below); they are to be used by the readers as starting points for student programmers or as learning tools for those just beginning spreadsheet programming. Figure 1 (Excel format) shows the formulas at the top of each range just as it would appear in the final spreadsheet. These formulas are to be **relatively** copied to cells below. (A relative copy is used so that each new version of the formula refers to the same relative cells in the spreadsheet). We have allowed a maximum of 10 cells in each of the necessary ranges for the calculation; however, by adding additional data rows the spreadsheet can be expanded to handle any number of data pairs. This requires, however, that cells whose relative values depend on the added data cells must themselves be added along with their relative formulas and that **absolute** cell referencing (see below) must be adjusted where necessary. The addition of an extra row in Microsoft Excel can be accomplished by selecting an entire row (clicking in the row header) and then choosing INSERT from the EDIT menu. To add additional cells in Quattro Pro, select the header BLOCK, then scroll down to INSERT, then select ROW. A box will appear, click on the OK button and a row is inserted.

The regions of the spreadsheet shown in Figure 1 that are available for expansion in this particular template are shown in gray. We have chosen this "size" of *xy* data capacity in order for the students to be able to see almost the entire spreadsheet on their computer screens at once. The technique for adding rows to a spreadsheet already containing formulas with relative referencing makes all necessary formulae adjustments in the relative references. This is one of the more powerful aspects of computer software of this type and is the reason these programs have revolutionized complex numerical calculations.

A few of the programmed formulas contain absolute cell references (e.g., cell C18 in Figure 1). Absolute cell references are designated in both the programs discussed here with a dollar sign ($) before the cell reference. Absolute cell references *do not* change if

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

	A	B	C	D	E	F
1	Col. Headers->	n	x	y	xy	x^2
2	Data point ----->	1			=IF(ISBLANK(C2),"",C2*D2)	=IF(ISBLANK(C2),"",C2^2)
3	Data point ----->	2				
4	Data point ----->	3				
5	Data point ----->	4				
6	Data point ----->	5				
7	Data point ----->	6				
8	Data point ----->	7				
9	Data point ----->	8				
10	Data point ----->	9				
11	Data point ----->	10				
12		# XY pairs	•x	•y	•xy	•x squared
13	Count function -->	=COUNT(C2:C11)	=SUM(C2:C11)	=SUM(D2:D11)	=SUM(E2:E11)	=SUM(F2:F11)
14						
15						
16			Averages of X			
17			av(x)=	av(y)=		
18			=C13/$B$13	=D13/$B$13		

FIGURE 1. SELECTED REGION OF THE STUDENT TEMPLATE SPREADSHEET. GRAY AREAS ARE FOR EXPANSION BY STUDENTS.

	A	B	C	D	E	F
1	Col. Headers ->	n	x	y	xy	x^2
2	Data point ----->	1	1.00	15520	15520.00	1.00
3	Data point ----->	2	2.00	31243	62486.00	4.00
4	Data point ----->	3	2.50	38575	96437.50	6.25
5	Data point ----->	4	5.00	74273	371365.00	25.00
6	Data point ----->	5	7.00	109569	766983.00	49.00
7	Data point ----->	6	7.50	114967	862252.50	56.25
8	Data point ----->	7	10.00	149631	1496310.00	100.00
9	Data point ----->	8				
10	Data point ----->	9				
11	Data point ----->	10				
12		# XY pairs	•x	•y	•xy	•x squared
13	Count function -->	7	35.00	533778.00	3671354.00	241.50

FIGURE 2. A SELECTED REGION OF THE SPREADSHEET CONTAINING DATA FROM THE STUDENT OCTANE CALIBRATION EXPERIMENT.

that formula is copied to another cell. Cell C18, in fact, contains a relative and an absolute cell reference. Therefore, when the formula in C18 (**=C13/$B$13**) was copied to cell D18 (to calculate the average value of the *y* data) it automatically appeared as **=D13/$B$13**. Additional programming points are included in Appendix B, which is a mini-primer for the spreadsheet programming necessary for this laboratory.

Two of the files available online with this paper (octane for Excel and octane.wb1 for Quattro Pro) contain student data (from the Fall of 1995). These spreadsheet files calculate

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

the regression line for seven xy data pairs from the chromatography experiment. The calculated result is an equation

- for a line with a y-intercept of 881 and slope of 15,074 ($r = 0.99936$), and one standard deviation of the slope is ± 240 and one standard deviation of the y-intercept is ± 1412.

Figure 2 is a selected region of the Excel spreadsheet from the file octane.

After the equation for the regression line has been generated by the spreadsheet, the calculated slope and y-intercept are used to generate 10 new xy pairs that are plotted as the LLS line (in this example). The graph files available online with this paper are the LLS line output from these data (octane graph or octgraph.wbl in Excel and Quattro Pro, respectively). Figure 3 is the Excel graph (octane graph). The plot seen here is for the calibration of peak areas (as determined by the gas chromatographic integrator) from seven injections of seven different octane standards (in hexane solvent) detected by a flame ionization detector. The range of the standards is from 1 to 10 parts per thousand of octane. In the plot, which shows regression line and calibration data on the same graph (Figure 3), the symbols for the points on the best-fit line have been turned off so that the points of the original experimental data can be more clearly examined.

Our experience has been that students can start this experiment with a set of “canned” calibration data that yields a known regression line. With this data they can usually program the spreadsheet in 2 to 3 hours, using the canned data to check for programming errors. Finally, students are required to analyze and plot three supplied sets of data and hand in the spreadsheet printout and plot for each. Annotation such as that shown in Figure 3 is required so that they must learn additional software functions.

One of the most common problems we have found in teaching this programming is that students don’t allow enough width in calculated spreadsheet cells for the number that is generated by the formula residing there. If the cells are not wide enough, then Excel will display a series of # symbols (Quattro Pro, a series of * symbols) to denote that the cell is not wide enough to display its numerical result. Simply widening (reformatting) that column solves the problem. Excel has a “Best-Fit” column width function that automatically makes selected columns as wide as necessary to accommodate the numbers residing there (under menu choices FORMAT/COLUMN WIDTH. . . /BEST FIT). In Quattro Pro

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

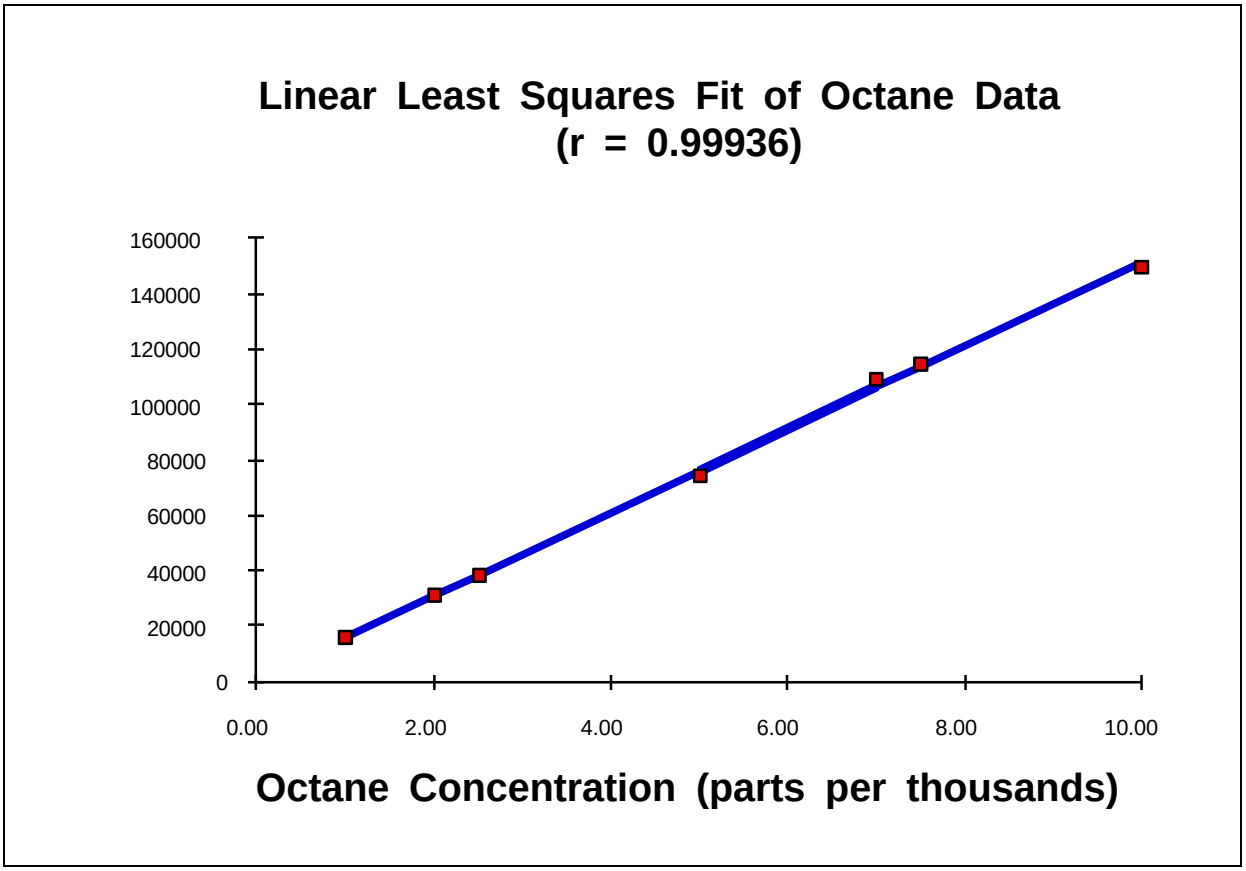

FIGURE 3. LLS LINE PLOT FROM OCTANE STUDENT CALIBRATION DATA. LINE IS FROM LLS EQUATION GENERATED DATA POINTS AND SQUARES REPRESENT THE EXPERIMENTAL DATA.

this function is available on the “SPEED BAR” by clicking the FIT button or under menu choice PROPERTY and sub choice COLUMN WIDTH.

Conclusions

In this online paper we have presented the “nuts and bolts” (especially in Appendix B) of programming a computer spreadsheet in order to calculate a least-squares line and to create a graph showing that line and the original experimental data. Our students use the spreadsheet that they program in this exercise for analyzing data from two other laboratory experiments performed later in the semester. They make a Beer’s Law plot using UV/visible spectroscopic data for the iron/*o*-phenanthroline complex and they use it to calibrate the GC data detailed in this article. The interpretation and limitations of LLS treatment is traditionally covered in Quantitative Analysis textbook [1], [2], [3], [4].

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

The role of the LLS treatment of data is as important now as ever, and it is more computationally accessible than in the past. At our university, drawing the "best looking line" though a set of calibration data is no longer acceptable, either in Quantitative Analysis or Instrumental Analysis courses. The advent of calculators and computer programs that perform this calculation easily does not decrease the need to understand the computational process required or the errors associated with the best-fit line.

ACKNOWLEDGMENT

The authors are grateful for a departmental grant from the Robert A. Welch Foundation and for discussions about the statistics included in Appendix A with Dr. Mark Carpenter. A reviewer's suggestions about the NAME function, included in Appendix B are appreciated.

REFERENCES

1. de Levie, R. *A Spreadsheet Workbook for Quantitative Analysis*; McGraw Hill: New York, 1992; pp. A2.1–A2.2.

2. Christian, S. D., Lane, E. H., and Garland, F. *J. Chem. Ed*. **1974**, *51*, 475–476.

3. Skoog, D. A., West, D. M., and Holler, F. J. *Fundamentals of Analytical Chemistry*; Saunders College Publishing,: Fort Worth, 1992; pp. 55–58.

4. Harris, D. C. *Quantitative Chemical Analysis*; Freeman: New York, 1995; pp. 71–81.

5. Irvin, J. A. and Quickenden, T. I. *J. Chem Ed*.; **1983**, *60*, 711–712.

6. Kalantar, A. H. *J. Chem. Ed*. **1987**; *64*, 28–29.

7. Tan, H. S. and Jones, W. E. *J. Chem. Ed*. **1989**, *66*, 650–651.

8. Ogren, P. J. and Norton, J. R. *J. Chem. Ed*. **1992**, *69*, A130–131.

9. Ott, R. L., *An Introduction to Statistical Methods and Data Analysis*, 4th ed.; Duxbury Press; Belmont, CA, 1993; pp. 491–494.

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

Appendix A

An assumption of linear regression is that x and y are linearly related according to the Following: $y = mx + b + \varepsilon$, where ε represents the random error (noise) with a mean of 0 and a constant variance [9]. Using this assumption of constant variance, we can estimate the standard deviation of the error (i.e., the standard deviation of the error about the regression line, S_ε), the standard deviation of the y-intercept, S_y, and the slope, S_m [3], [9]. These standard deviations are calculated as follows. Here we assume that the sample S_ε approximates the population S_ε.

$$S_\varepsilon = \sqrt{\frac{\Sigma(y_i - mx_i - b)^2}{n-2}}$$

$$S_b = \sqrt{\frac{S_\varepsilon^2 \Sigma(x_i^2)}{n\Sigma(x_i^2) - (\Sigma x_i)^2}}$$

$$S_m = \sqrt{\frac{S_\varepsilon^2 n}{n\Sigma(x_i^2) - (\Sigma x_i)^2}}$$

Two additional spreadsheet files (octane error and eroctane.wb1, in Excel and Quattro Pro, respectively) contain the octane student data and calculations included in octane and octane.wb1, plus the standard deviation equations described above.

REFERENCES

9. Ott, R. L., *An Introduction to Statistical Methods and Data Analysis*, 4th ed.; Duxbury Press; Belmont, CA, 1993; pp. 491–494.

Appendix B

Absolute and relative cell referencing was discussed in the body of this article. Additionally, a few other spreadsheet commands are covered here. All of the commands mentioned here are, of course, covered in the Excel and Quattro Pro manuals.

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

COUNT formula

In order to automatically keep count of the number of xy data pairs that have been entered (and used later for calculating averages of several columns) we have used the spreadsheets' count function. In Excel this formula looks like the following: **=COUNT(C2:C11)**. The equal sign starts the formula; the count function is the word COUNT and the range of cells in parentheses are the cells that will be counted. In this example, the count function will be incremented by one for each cell in the range C2 through C11 containing the x_i data column in our example. Nonnumerical data do not increment the counter. We have arbitrarily chosen the x data column; counting the y data column entries would be equally valid since the data come in pairs.

In Quattro Pro this function looks like: **@COUNT(C2..C11)** where the only difference is that the formula is started by @ instead of =.

IF-THEN formula

To generalize the spreadsheet so that the regression line is automatically recalculated when a new xy data set is entered and to ensure that after the initial (correct) programming no additional formula copying or manipulation is required, we have used functions that check to see if an entry has been placed in a source cell and then act based on the result. If an entry *is* found then that entry is used in a calculation and the result appears in the cell containing the IF-THEN formula. If no number appears in the source cell then the cell containing the formula will appear blank.

In Excel this formula is: **=IF(ISBLANK(C9),“ ”, D9*C9)**. The formula checks to see if there is a value in cell C9. If not (i.e., C9 is blank) the formula produces a space in the cell containing this formula. If, however, there is a value in cell C9, then the cell containing this formula will show the product of cell D9 and cell C9 (to calculate $x_i y_i$, for instance). Obviously the last entry in the ISBLANK formula could be any calculation that should occur if the source cell contains a value. The function should be used in all the cells in the spreadsheet where calculations are to be carried out (only) if a source cell contains a value and left blank if not. We have used this formula in the files available with this paper in all data series that come after the xy data set.

In Quattro Pro the analogous function is: **@IF(C9=“”, “”, C9*D9)**. Instead of having a dedicated function that, by default, checks for a blank space, this function checks to see

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

if the target cell (again C9) contains "" (i.e., nothing) and if it does the result is a space. If it contains a value then the result will be C9 multiplied by D9.

All of our IF-THEN formulas contain a reference to the *x* data column in order to decide whether or not to perform the argument's calculation. This is because if there is not an *xy* data point then the subsequent calculations are unnecessary, and because the later formulas depend on the sum of a column containing these formulas, the cell must have either nothing or a number produced by its formula. Otherwise, any cells referencing this cell will generate an error.

NAME Function

We have not used the name function in the spreadsheet programming described here, but it is an alternative, and perhaps more intuitive, method of referring to cell addresses. In both Excel and Quattro Pro, a range of cells (or a single cell) can be selected and assigned a name using the protocol of the program: for instance, all of the data points in the *x* data column can be selected and then named **X** (using menu choice FORMULA, subchoice DEFINE NAME. . . in Excel or header BLOCK, NAMES, DEFINE in Quattro Pro). Subsequently, the name **X** can then be used in a cell formula in place of the relative cell reference. Likewise, using this same function to name the cell containing the calculated *y*-intercept and slope eliminates the need to adjust the absolute cell referencing if the spreadsheet data columns are expanded later. Most importantly, calculations using named cell ranges may be clearer and more intuitive to the beginning programmer.

Plotting

The Excel graph included as file "octane graph" (and shown in Figure 3) gives one example of how to plot the calibration data and the best-fit linear-least-squares line. Note that this file can be opened without the associated spreadsheet file being present on the disk (not so of Quattro Pro, see below).

As stated in the article body, the data *symbols* or *markers* have been turned off so that all that is left is the regression line. Secondly, the calibration data's *line* has been turned off so that all that is displayed is its markers. In this way the plot most closely mimics linear-least-squares plots generated by most dedicated software programs.

ISSN 1430-4171
http://journals.springer-ny.com/chedr
S 1430-4171 (96) 01012-6

The data used to plot the least-squares line on the graph in Figure 3 is a single *x* series (Series is Excel's term for a column of data, B2:B8 in our spreadsheet example) and two *y* series, one from the calculated best-fit *y* data (B21:B27) and the other from the original calibration data (D2:D8). The choice of this method of plotting is an effort to be consistent with the plotting method of Quattro Pro.

Initially, we programmed the Microsoft Excel spreadsheet using a different set of *x*-coordinates to calculate the matching *y*-coordinates of the best-fit line, and these *xy* pairs were used to plot the line. We then plotted the two data sets, one the best-fit *xy* data and the other the experimental calibration data, on the same graph. This worked fine in Excel. Any *x* values could have been used to generate corresponding *y*-coordinates that, when plotted, would give the correct best-fit line. This method, however, cannot be used in Quattro Pro. It can associate *y* data sets with only a single set of *x* values; thus, by using the same *x* values that were used experimentally to calculate the corresponding best-fit *y* values, the plot can be generated similarly in both programs.

Finally, Quattro Pro does not save graphs as separate files as does Microsoft Excel. For this reason, the Quattro Pro graph of the octane data and best-fit line is actually included in a spreadsheet file (octgraph.wb1) that has been saved in such a manner that the graph is placed on top of the spreadsheet containing the programmed equations that generated the graph. The reason that only the graph appears, not the underlying equations, *xy* data, etc., is because we have made the spreadsheet's text color white. In order to reveal the underlying spreadsheet, select all cells, choose "Property"/"Current Object"/"Text Color" from the menu, and then choose any color, except white.