

THE POWER OF SPREADSHEETS

in a Mass and Energy Balances Course

MICHAEL MISOVICH, KARYN BIASCA
*University of Wisconsin
 Stevens Point, WI 54481*

The use of computers for design and analysis in chemical engineering applications has flourished in the past decade, and most chemical engineering curricula include a computing language requirement, such as FORTRAN, BASIC, or PASCAL, usually taken in the freshman or sophomore year. Upper level courses often include instruction in the use of software packages for process simulation and design.

We feel, however, that the use of spreadsheets for engineering calculations has not been widely enough integrated across the curriculum. We feel that this kind of instruction is particularly important since graduating engineers may not have access to sophisticated software packages or programming languages, but will almost certainly have spreadsheet software available for their use. Although the structure and use of spreadsheets is quite different from programming languages, many of the same kind of problems can be solved by either method.

Recent articles have appeared which described the use of spreadsheets for chemical engineering calculations^[1] and their use in a senior design class.^[2] We have found it fruitful to expose students to spreadsheets at the earliest possible point in the curriculum—in the sophomore mass and energy balances class. Because of the relative ease in learning spreadsheet software, we find that the investment of a few hours of class time at this point allows students to work more meaningful problems and to examine problems in greater detail.

ADVANTAGES OF SPREADSHEETS

Many engineering students find that using FORTRAN or a similar language makes a problem more difficult to solve because the bulk of their effort must be centered on the details of programming syntax. As instructors, we sometimes compensate for this by not making the engineering part of the problem too difficult or too time-consuming. Students then get

the message, "I know this is an easy problem, but look how hard it becomes when using the computer," and as a result, they may subsequently avoid using the computer and avoid solving complicated problems for which hand-solution is not feasible.

Using spreadsheets alleviates some of the problems that students encounter with traditional programming languages. The syntax of spreadsheets essentially consists of algebraic formulas and functions, concepts which are familiar from elementary algebra. On the other hand, programming languages require the mastery or memorization of numerous keywords, punctuation marks, and special symbols, which in most cases must be combined by using rigid rules. So it is not surprising that students can learn to do meaningful calculations much more quickly by using a spreadsheet than they can with a programming language.

Spreadsheets also help reduce the proliferation of syntax errors and logical errors. If a syntactically incorrect formula is entered, the programmer is immediately notified of the error and can correct it before continuing. Since the numerical result of every formula is displayed after it is entered, grossly incorrect results can be seen immediately, and logical errors in formulas can be corrected before being propagated through the remaining calculation steps.

Michael J. Misovich is an assistant professor in the paper science department at the University of Wisconsin-Stevens Point. He holds a PhD in chemical engineering and an MS in computer science from Michigan State University.



Karyn L. Biasca is an assistant professor in the paper science department at the University of Wisconsin-Stevens Point. She holds a PhD from the Institute of Paper Chemistry and a BS in chemical engineering from the University of California at Los Angeles.

© Copyright ChE Division ASEE 1991

Chemical Engineering Education

In typical programming languages, such errors may be difficult to debug without adding numerous PRINT or WRITE statements which will list results of all intermediate calculations leading up to the wrong answer.

It is easy for students to document their spreadsheet work because of the convenience of entering text labels next to numeric cells. When programming languages are used, students often avoid such documenting features as comments and PRINT statements to label output because of the extra effort needed to include them.

Using a spreadsheet also helps students to efficiently organize their calculations. The grid structure facilitates creation and organization of one- and two-dimensional tables. Contrast this to FORTRAN, where creating a two-dimensional table requires the correct nesting of an implied DO loop inside an ordinary DO loop, manipulating array subscripts in the proper order, and correct interaction of the implied DO loop with its FORMAT statement.

Flexibility of data input is more apparent in a spreadsheet, since results can be recalculated when any cell is changed, at any time. In programming languages, only variables listed in READ or INPUT statements can be changed while the program is being executed. Students may circumvent this by using an assignment statement: an example of this in BASIC would be "LET TEMP = 298" rather than "INPUT TEMP". This makes testing several cases or ranges of values for independent variables much less convenient.

Since spreadsheet software includes graphics capability, students can generate graphical results with ease. Displaying the results of calculations in a chart or graph often provides additional insight into the physical significance of their calculations. When using programming languages, such graphics may be generated only by special subroutines or procedure calls which are not familiar to many students.

The human element of programming has been studied in the fields of software engineering and structured programming,^[3] and it is believed that excessive flexibility in program design leads to programs which are complicated to understand and difficult to debug. Spreadsheet programming offers fewer of these pitfalls because it works with an inherently structured format. We believe that the use of spreadsheet software aids students in organizing their work on problems done by hand. The rows and columns of a spreadsheet encourage the habit of

setting up tables to solve mass and energy balance problems. This approach is often helpful (particularly to novices) but is not apparent when problems are solved by writing conventional computer programs.

One criticism of spreadsheet software is that it lacks the power of IF-THEN-ELSE and DO statements found in programming languages. Yet most spreadsheets *do* contain functions which implement

We feel . . . the use of spreadsheets for engineering calculations has not been widely enough integrated across the curriculum. . . . students can learn to do meaningful calculations much more quickly by using a spreadsheet. . .

IF-THEN-ELSE logic, and most of them support iteration (which is probably the most common use of repetition logic in engineering calculations) by circular recalculation. Most also support some type of "macro," "program," or "project" processing, which constitutes a programming language for manipulating the spreadsheet. Examples of this are given in Reference 4.

When the above factors are considered, the functionality of spreadsheet software is comparable to programming languages for most types of engineering calculations. Actually, however, we do not need to make use of these advanced features in the mass and energy balances class since this type of logic is not usually necessary in balance calculations. A straightforward accounting of inputs and outputs around a system boundary usually suffices, and spreadsheets are the ideal software for such calculations.

Spreadsheets are not a substitute for modular simulation packages such as Aspenplus.^[5] For complicated, multiple-unit balances, these packages are preferred because the balance calculations and physical data are built in. To students who are solving simpler problems and are learning to solve such problems for the first time, the overhead associated with simulation packages makes the problem seem more difficult to solve. Also, students do not practice setting up and solving balances when the package does it for them.

SPREADSHEET INSTRUCTION

We present several spreadsheet computing problems during the mass and energy balances class. Typically, students entering this class have taken FORTRAN as freshmen or sophomores, but few have

had experience with spreadsheets. Since spreadsheets are a powerful and convenient tool which will be of value to the student throughout the entire chemical engineering curriculum, we felt that we should offer instruction at the earliest possible time. That is why we chose this particular class as the vehicle for spreadsheet instruction. As a result, instructors in subsequent classes who want to set up computing problems will find that they have students who are familiar with spreadsheets.

One of the principal advantages of spreadsheet computing is that it is simple to learn. In a two-hour, computing lab period, students can learn enough to complete a mass balance for a combustion process in which an independent variable such as the percent-of-excess air or percent-of-incomplete combustion is varied. They can construct tables and graphs illustrating these results.

Many mass and energy balance problems can be completed with only knowledge of a small set of operating concepts and commands. The concept of cell addresses is vital, as is the distinction between numeric data, text data, and formulas. Commands for formatting cells, editing existing cells, and copying cells are needed. For the copying operation, the distinction between absolute and relative cell addresses is important. Although some spreadsheets refer to this as "copy without adjustment" and "copy with adjustment," all perform the same function—they hold either the row or column number of a cell fixed during copying. This set of commands is sufficient to create and edit a spreadsheet and to display results on the monitor. In addition, commands to produce printed results and to save and reload previous work are needed. Finally, commands for producing graphics are used to identify the coordinates to be graphed, to produce labels and titles, and to display or print the graph.

Students are encouraged to obey a few customary rules of spreadsheet usage in our class. We ask that all numeric input data be placed at the top and all formulas be placed at the bottom of the spreadsheet, with the sections separated by a visible line. All values, whether input data or calculated output, should be labeled. Circular (or forward) references, in which a formula contains a cell which is in a later row or column, should be avoided unless iteration is specifically intended. These may lead to errors when input data are changed because the formula will calculate results using cells which themselves have not been recalculated based on the new data.

We expect students to work substantially through a combustion mass balance problem, applying these spreadsheet concepts, in a two-hour lab period. Each step is explained in a written handout, and the instructors are available to give immediate assistance to any student who becomes confused.

The first time this introductory exercise was given, in 1985, students completed a follow-up survey which included the question, "Would you find solving problems like the one we did easier using a spreadsheet, or by writing a computer program in FORTRAN, BASIC, PASCAL, *etc.*?" Of forty-one students who responded, 80% preferred the spreadsheet, 17% did not know, and only 3% (one student) preferred using a programming language. Some of the student comments we received were:

- **Using a spreadsheet keeps the problem more organized and actually saves time.**
- **It's easier in spreadsheet because it simplifies the process of solving. The other languages tend to complicate the material balance.**
- **In the spreadsheet, all data was visible at all times and the data was set up in a nice tabulated form that was easily changed.**
- **Mistakes were easier to correct**

Since this first experience, we have refined the exercise by improving handouts and becoming acclimated to the potential problems caused by a relatively large student group using a public computing lab.

ASSIGNMENT EXAMPLES

We typically assign from five to eight spreadsheet problems during a quarter or semester class. These problems are given in sequence with the material as it is discussed in lecture. They also coordinate well with the presentation of subject matter in textbooks we have used.^{16,71} Copies of these assignments are available from the authors upon request.

The combustion mass balance problem described above is given first. The goal is to calculate wet and dry analysis of a flue gas. Typical variables include the empirical formula of the compound burned, degree of incomplete combustion, and percentage of excess air supplied. Later, in another problem, this same spreadsheet is expanded to include gas concepts such as volumetric flow rate, partial pressure, vapor pressure, humidity, and dew point. After energy balances have been studied, the same problem is expanded a third time to include temperature and enthalpy data, with the goal of calculating adiabatic

combustion temperature. This three-problem series is always used.

A fourth problem is used, when real gases are studied, to demonstrate the link between equations of state and generalized compressibility charts. To students, these methods may seem entirely dissimilar. Using a spreadsheet, several constant temperature or constant volume curves of a compressibility chart are plotted with PVT data generated from an equation of state. When students compare their own curves to those shown in the textbook charts, the reaction is often one of amazement over the striking similarity of the results. A sample of typical output from this assignment is given in Figure 1.

The graphics capability of spreadsheets is also applied in two other optional assignments. The first of these problems asks the students to generate an enthalpy-concentration diagram for a binary system (such as water-sulfuric acid) using tabulated heat capacity and enthalpy of solution data. The second problem is the construction of a psychrometric diagram from equations describing vapor pressure and enthalpy. As with the compressibility diagram assignment, these assignments tie together the algebraic, tabular, and graphical techniques of problem solving. Students see that similar results can be obtained by using different methods. A recent article^[8] recommends using "computer friendly" techniques in place of, or in addition to, graphical techniques whenever possible; the above assignments provide examples of this.

Other problems that are sometimes used are 1) the generation of enthalpy of vaporization data as a function of temperature from various forms of the Clausius-Clapeyron equation or the Watson correla-

tion, and 2) generation of an enthalpy table (like the steam table) for a substance other than water. These problems demonstrate the correspondence between equation-oriented and tabular approaches for estimating enthalpies.

We have taken a multiple-effect evaporator problem from a textbook.^[7] This problem is readily adaptable to a spreadsheet solution as the problem statement requests that a table of properties be filled in for each effect. Each effect obeys similar equations, so a copying operation can be used to fill in the entire table once the equations for one effect have been entered. The manipulation of one cell containing an independent variable (the vapor flow from the last effect) allows recalculation of the entire table. Students adjust this value by trial-and-error until the desired liquid composition leaving the last effect is generated. This problem demonstrates that trial-and-error can be a practical solution technique for complicated problems when the computer is performing the calculations. Students may at first resist using trial-and-error because they have been taught it is not mathematically "elegant" and can be labor-intensive when done by hand.

Students have successfully solved the problems described here using several spreadsheet packages, including Lotus 1-2-3,^[9] Supercalc3 and Supercalc4,^[10] and the Smart Spreadsheet,^[11] on several types of IBM and IBM-compatible microcomputers. The only major difference between these packages is the adjustment of absolute and relative cell references in copying operations. Other differences between these spreadsheets are usually immaterial to students in our class.

Once a spreadsheet has been designed, students can use it to answer questions or to gain additional understanding or insight into the problem. For example, we typically require students to solve the adiabatic combustion temperature problem for two different compounds, and then they compare the results and report on their findings in a short memorandum similar to one described in an article by McKean and Hanzevack.^[12] The class can then pool its results and discuss questions such as, "What chemical characteristics appear to lead to higher combustion temperatures for fuels?" or "Why do we use carbon (coal) and methane (natural gas) as fuels when their combustion temperatures are among the lowest of the compounds under the conditions we studied?" Similar discussion can be facilitated for many of the other problems by assigning different combinations of independent variables to different

Continued on page 52.

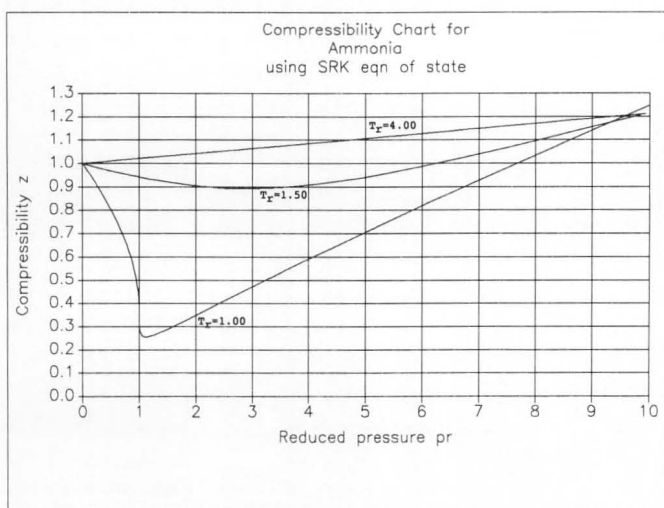


FIGURE 1. Sample compressibility chart.

$$r_1 = 0.99744 \quad r_2 = 0.56615 \quad (27)$$

Applying Eq. (22) to above the feed position, $n = 1$, $y_1 = 0.995$, inserting the appropriate values for A, B, m, and b from Eqs. (23) and (24) with reference to Eqs. (10) and (11), and using $r_1 = 0.75130$ and $r_2 = 0.40920$, we obtain y_{n+1} by substituting the feed composition into the operating line as the feed is introduced at its bubble point. Thus $y_{n+1} = 0.549$ (or 0.553 using the lower operating line).

Applying the values

$$\begin{array}{ll} r_1 = 0.75130 & r_2 = 0.40920 \\ y_1 = 0.995 & y_{p+1} = 0.549 \\ B = 0.59 & m = 0.75 \\ A = 0.41 & b = 0.249 \end{array}$$

results in 9.54 plates above feed position.

Below the feed position, Eq. (22) may be applied by taking $n = 9.54$, i.e., $y_{9.54} = 0.549$. We obtain $y_{p+9.54}$ by substituting values for A, B, m, and b from Eqs. (23) and (25), and using $r_1 = 0.99744$ and $r_2 = 0.56615$. We obtain $y_{p+9.54}$ by substituting $x = 0.005$ into the equilibrium line to give $y_{p+9.54} = 0.0121$. Thus

$$\begin{array}{ll} r_1 = 0.99744 & r_2 = 0.56615 \\ y_{9.54} = 0.549 & y_{p+9.54} = 0.0121 \\ B = 0.59 & m = 1.3773 \\ A = 0.41 & b = 0.001886 \end{array}$$

These values, when substituted into Eq. (22) yield 9.91 plates below feed position.

CONCLUSIONS

The binary distillation problem considered by Amundson was re-examined, and a simpler method involving powers of matrices has been given and an explicit solution obtained. This approach is suitable for use in engineering mathematics or separation processes courses to illustrate the application of matrices to engineering problems.

ACKNOWLEDGEMENTS

The author is grateful to his colleague, Kevin Free, for improvements in the clarity of this paper.

REFERENCES

1. Amundson, N., "Application of Matrices and Finite Difference Equations to Binary Distillation," *Trans. AIChE*, **42**, 939 (1946) □

POWER OF SPREADSHEETS

Continued from page 49.

students. Insight into the relative importance of variables and sensitivity of results to changes in the input can be gathered from such an exercise. The ease of changing input data also allows instructors to efficiently check calculations made with different combinations of independent variables.

CONCLUSIONS

Our experience with spreadsheet computing has proved to us that it is feasible to provide instruction on spreadsheet use as part of the mass and energy balances class. Within a time-frame of approximately two hours, students can learn sufficient fundamentals to use spreadsheets as a tool for solving a variety of problems in the class. After solving five to eight problems, most of them have enough confidence and experience to apply the techniques in future engineering classes.

The use of spreadsheets also encourages organization in problem solving which hopefully will carry through to the student's non-computer work. The flexibility and convenience of spreadsheets allows students to solve more meaningful problems and to examine the solutions in detail by manipulating independent variables to determine their effect. The built-in graphics capability also helps to tie together graphical and algebraic solution techniques when such alternate methods exist for a given problem.

REFERENCES

1. Rosen, E.M., and R.N. Adams, "A Review of Spreadsheet Usage in Chemical Engineering Calculations," *Computers and Chem. Engg.*, **11** (6), 723 (1987)
2. Grulke, E.A., "Using Spreadsheets for Teaching Design," *Chem. Eng. Ed.*, **20**, 128 (1986)
3. Dijkstra, E.W., *A Discipline of Programming*, Prentice-Hall Inc., Englewood Cliffs, NJ (1976)
4. Rosen, E.M., "The Use of Lotus 1-2-3 Macros in Engineering Calculations," *Chem. Eng. Ed.*, **24**, 100 (1990)
5. Aspenplus, Aspen Technology Corporation, Cambridge, MA
6. Himmelblau, D.M., *Basic Principles and Calculations in Chemical Engineering*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1982)
7. Felder, R.M., and R.W. Rousseau, *Elementary Principles of Chemical Processes*, John Wiley & Sons, New York (1986)
8. Wankat, P.C., "What Will We Remove From the Curriculum to Make Room for X?" *Chem. Eng. Ed.*, **21**, 72 (1987)
9. Lotus 1-2-3, Lotus Development Corporation, Cambridge, MA
10. Supercalc, Computer Associates International, San Jose, CA
11. Smartware, Informix Software Inc., Lenexa, KS
12. McKean, R.A., and E.L. Hanzevack, "The Heart of the Matter: The Engineer's Essential One-Page Memo," *Chem. Eng. Ed.*, **23**, 102 (1989) □