COMPUTER SCIENCE OR SPREADSHEET ENGINEERING?

An Excel/VBA-Based Programming and Problem Solving Course

DANIEL G. CORONELL
Rose-Hulman Institute of Technology • Terre Haute, IN 47803

Over the past two decades, chemical engineering practice has been profoundly influenced by advances in computer hardware and software, stimulating debate within the academic community on how students should be prepared for computer applications in the "real world." At the crux of this debate is the relative importance of including a traditional introductory computer science course in the chemical engineering curriculum. Without question, in the "old days" the pathway for applying computers to the solution of engineering problems was to write your own program from scratch, usually in Fortran.

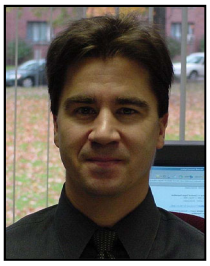
Today, industry is much less likely to engage their engineers with such tasks. The expectation is that commercial software vendors will supply them with user-friendly software packages that require little or no programming skills. A recent survey[1] by CACHE indicated that computing in the workplace for entry-level chemical engineers is clearly on the rise (over two-thirds of the approximately 300 respondents spent at least one-half of their workday at their computer). It was found that most of the time spent working on the computer involved user-friendly commercial software packages, with the most common application being Microsoft Excel. Nearly three-quarters of the respondents were not expected by their employers to be competent in any programming language. The most common programming language being used, and the one most highly recommended for inclusion in the chemical engineering curricula, was Visual Basic.

Based on these results, it might be argued that graduating chemical engineers would be more suitably equipped to contribute in an industrial setting if they were taught how to effectively use Excel rather than how to write a computer program in a language they may never use again. The numerous books,[2-5] trade journal articles,[6-8] software vendors,[9-12] and consultants[13-15] that demonstrate the use of Excel in engineering analyses underscore this point. This notion, however, overlooks the value of learning how to logically formulate a problem-solving strategy that is inherent in any programming course. Moreover, it omits the necessary exposure to programming concepts (*e.g.,* loops, decision constructs, *etc.*) for the fraction of students who may be required to do some type of programming in an R&D setting or in graduate school.

This paper describes a compromise approach that combines instruction on the use of Excel as well as computer programming concepts by way of Excel's macro programming language, Visual Basic for Applications (VBA). The benefit to students is that they can learn the practical aspects of "spreadsheet engineering" as well as the more generally applicable concepts of computer programming. Additionally, they gain a clearer understanding of how the course material applies to their future profession since the course is taught within the chemical engineering department. The benefit to the instructor is the ability to consolidate the presentation of course material through the use of a single software package. This paper describes the format and content of a freshman-level course that has been designed to replace the more traditional introductory computer science course.

COURSE FORMAT

Programming and Computation for Chemical Engineers is a two-credit-hour course that chemical engineering majors at Rose-Hulman Institute of Technology are required to take in the spring quarter of their freshman year. The class meets



Dan Coronell *received his BS from the University of Illinois-Urbana and his PhD from the Massachusetts Institute of Technology, both in chemical engineering. After graduation he worked in the chemical, semiconductor, and engineering software industries for over nine years before joining the faculty at Rose-Hulman Institute of Technology, where he is presently Associate Professor.*

two times per week for fifty-minute periods over a ten-week quarter. At this point in the curriculum, students have typically completed two quarters of calculus and chemistry and one quarter of physics. They are also concurrently enrolled in a freshman-level design class that introduces them to many concepts of importance to chemical engineers[16]. The early introduction of chemical engineering concepts into their curriculum provided by the two courses is beneficial to the students, as will be discussed below.

Prior to the first meeting of this newly redesigned course, a survey was conducted to assess the level of expertise in using Excel and experience with any programming language. Approximately two-thirds of the 66 students that were originally registered for the two sections responded to the survey. The first part of the survey asked the students to select one of five different categories that best characterized their ability to use Excel. The selection options included

- Power user
- Pretty comfortable using it to process data and make plots
- Have used it before several times and know the basics
- Have only started using it since my freshman year
- Have never used it before

The results are summarized in Table 1. While all of the respondents had used Excel to some extent, most of the course material consisted of techniques and applications that the students had never been exposed to in the past.

The second part of the survey asked students to identify any programming languages they had previously learned in coursework or through work experience. As can be seen in Table 2 below, two-thirds of the respondents had no previous programming experience. Note that a few respondents had experience in more than one type of programming language.

The classroom instructional technology at Rose-Hulman greatly facilitated the execution of this type of course. Every classroom is equipped with a laptop computer projector and wireless network capabilities. In addition, each student at Rose-Hulman is issued a laptop computer at the beginning of their freshman year. The students were required to bring their laptop computers to class each day. A typical 50-minute class period was discretized into 20 minutes of traditional lecture, 20 minutes of computer laboratory, and 10 minutes of discussion and reflection.

TABLE 1
Survey of Students' Level of Expertise Using Excel

	Power User	*Pretty Comfortable*	*Know Basics*	*Just Started*	*Never Used Before*
# of Respondents	2	17	13	10	0

TABLE 2
Survey of Students' Prior Programming Experience

	Visual Basic	*C/C++ C#*	*Java HTML*	*Pascal*	*Matlab*	*None*
# of Respondents	10	9	5	1	1	28

The initial lecture period would usually include an instructor-led example problem with students following along on their laptops, followed by a computer lab assignment on a problem related to the lecture topic. The students were free to work together and to ask questions during this time. The last few minutes of class were used to obtain closure on the subject matter where the computer lab solution would be provided, the relevance of the material would be reinforced, and any remaining questions would be answered. All in-class quizzes and homework assignments were submitted electronically as Excel workbooks.

While most of the students had not yet taken any core chemical engineering courses, every opportunity was taken to expose the students to the kinds of problems they would see later in the curriculum. This served to benefit the students in several ways. First, they became more engaged in learning the programming and the problem-solving concepts when it was demonstrated that these fundamentals could be applied to chemical engineering-related problems. This was true in spite of the fact that they possessed only a cursory understanding of the underlying fundamentals at this point in their education. Another benefit was that, early in the curriculum, students were exposed to a sample of what the chemical engineering profession entails. It was observed that many of the students were interested in chemical engineering for reasons ranging from the desire to follow the path of a family member or close friend to a desire to have a good paying job when they graduated. Approximately 5% (3/66) of them ended up changing their major after learning more about the chemical engineering profession. In the following section, the specific learning objectives and content of the course are described.

COURSE OBJECTIVES AND CONTENT

In a broader perspective, the objective of *Programming and Computation for Chemical Engineers* is to begin the process of introducing the computer as an engineering problem-solving tool. As described in the preceding section, the approach to satisfying this objective relies upon active learning, relevant example applications, and modern classroom instructional technology. This high-level objective of the newly redesigned course was refined into the specific learning objectives of

- Becoming proficient at using Excel to perform scientific and engineering calculations and graphical analysis.
- Understanding the essential elements of structured and object-oriented programming as it applies to VBA.
- Being able to construct customized VBA-based spreadsheet functions to enhance the engineering problem-solving capabilities of Excel.

The first one-quarter of the course consisted of formal instruction on spreadsheet techniques and tools. This is illustrated in Figure 1a, where the students' progression of learning begins with basic operations in a worksheet cell and progresses outward to increasingly complex worksheet operations. The remaining three-quarters of the course was devoted to instruction on VBA programming in Excel. Since VBA is a separate application that is integrated into the Excel environment, this involved two distinct aspects—learning the VBA programming language and learning how to interface with Excel from a VBA program. The latter required the students to work with Excel's so-called Object Model, a heirarchical, object-oriented interface to the Excel application that facilitates manipulation of all the elements of an Excel workbook.

A synopsis of the programming elements of VBA that were included in the course is shown in Figure 1b. VBA contains many of the same elements that are common to other programming languages—variables, loops, decision constructs, *etc*. Thus, the students obtain a conceptual understanding that enables them to more easily learn a different progamming language, such as C++ or Java. This was an important consideration since some of the students will continue their education in graduate school where they may be involved in computational research requiring programming skills in other languages.

EXAMPLE APPLICATIONS

In the preceding section, the general features of Excel spreadsheets and VBA programs that the students were exposed to, and which underly the course learning objectives, were described. The students developed their skills at using these tools by applying them to a number of engineering-related problems. The types of applications that were included and the specific example problem they were asked to solve are summarized in Table 3.

Most of the applications were first explored using a spreadsheet-only approach, then subsequently implemented in a VBA program. It is again emphasized that while the students did not possess a deep understanding of how the design equations for a particular application were derived, their appreciation for the usefulness of the computer skills they were acquiring was nonetheless heightened. Moreover, they finished the course with a better understanding of what was ahead of them in the chemical engineering curriculum.

One of the applications listed in Table 3 is the numerical solution of ordinary differential equations (ODEs). The students learned how to solve ODEs using both the explicit Euler method as well as the 4th-order Runge-Kutta method. As an example, the following differential equation representing the draining of a cylindrical tank was numerically solved by applying the 4th-order Runge-Kutta method, using both the spreadsheet and VBA program implementations.

$$\frac{dh}{dt} = -\left(\frac{d_{hole}}{d_{tank}}\right)^2 \sqrt{2gh}$$

In this equation, h is the height of fluid in the tank, t is the cumulative draining time, d_{hole} is the diameter of the drain hole located at the bottom of the tank, d_{tank} is the tank diameter, and g is the gravitational acceleration constant. This equation is widely known as Torricelli's formula, and possesses an exact solution. This enabled the students to also explore the concepts of integration step size and associated error, as well.

The two implementations are shown in Figure 2 below. The students solved this problem using the spreadsheet implementation early in the quarter, and subsequently revisited the problem after learning how to create customized VBA function procedures. This helped them to appreciate the advantages of the VBA approach, including the conciseness of the implementation

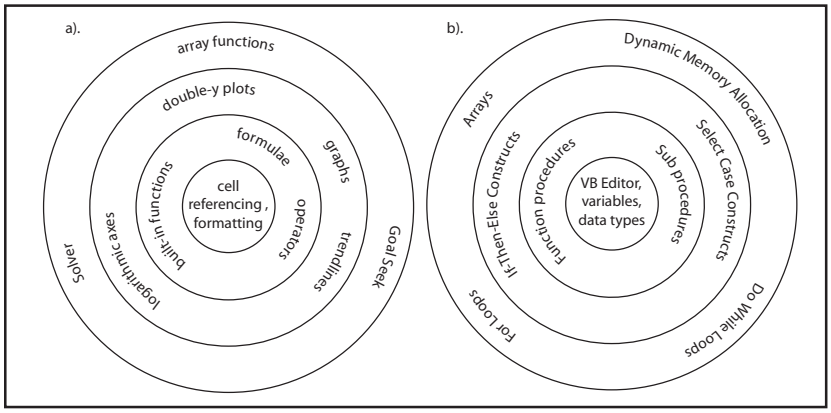


Figure 1. *Schematic illustrating sequence of topics pertaining to spreadsheet (a) and VBA (b) instruction.*

TABLE 3
Applications of Computer Skills Included in Course

Application	*Implementation*	*Example*
Engineering formula involving transcendental functions	Excel VBA	Pressure dynamics and release rate of choked flow from a high-pressure gas cylinder
Parameter estimation	Excel VBA	Determination of first-order rate constant from experimental data using the integral method
Solution of linear systems of algebraic equations	Excel	Steady-state material balance
Solution of nonlinear algebraic equations	Excel VBA	Determination of friction factor using the Colebrook equation
Numerical integration	Excel VBA	Sizing of a gas-liquid scrubber
Numerical solution of ordinary differential equations	Excel	Draining of a tank using Torricelli's formula

and the ability to reuse the function to solve a different problem by simply redefining the ODE in the VBA function. Additionally, the VBA implementation reduces the possibility of introducing a typographical error since the Runge-Kutta terms do not have to be retyped for each problem.

SUMMARY AND CONCLUDING REMARKS

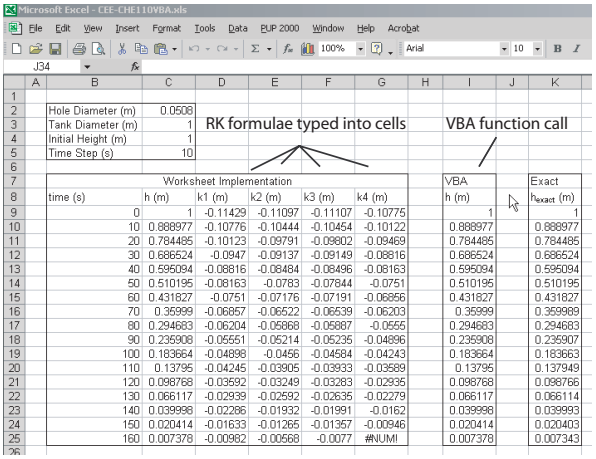
A newly redesigned freshman programming course for chemical engineers that supplants the traditional introductory computer science course has been described in this paper. The course focuses on the use of Excel spreadsheet and VBA programming techniques to solve engineering-related problems in response to the needs of industry as unveiled in a recent CACHE survey. The course also serves as a metric for students to assess their interest and aptitude for the chemical engineering profession at an earlier point in their curriculum than previously allowed.

Some remaining issues will have to be addressed, one of which includes the lack of suitable textbooks that relate to the course content. Many textbooks on Excel and VBA programming are available, but no textbook could be identified that focused on engineering-related applications. The students responded quite positively to learning programming skills within the context of solving engineering-related problems. A textbook that is tailored to such a class would greatly facilitate the continued offering of the course. Additionally, in order for the students to sustain and leverage the skills obtained in the course, a department-wide involvement to incorporate Excel/VBA problem-solving methods where applicable into their higher-level engineering courses must be initiated and sustained. The familiar saying, "Use it or lose it!" applies here.

It is also important to acknowledge that while spreadsheets and Visual Basic programs are useful for solving many relatively routine engineering problems as illustrated in the preceding section, some problems require more sophisticated computational tools. This becomes apparent to the students later in the curriculum as they take courses in fluid mechanics, transport phenomena, thermodynamics, reactor engineering, and design. Here they will learn about software packages designed to perform computational fluid dynamics calculations, predict detailed fluid properties, optimize process flowsheets, and more. Thus, the *Programming and Computation for Chemical Engineers* course represents the commencement of their education in using the computer as an engineering tool. The real value of the approach outlined here is, perhaps, that the students can more clearly and immediately see that computers can be programmed to efficiently solve chemical engineering problems.

a).

Excel worksheet for implementing spreadsheet and VBA-based solution to ODE.



	Hole Diameter (m)	0.0508								
	Tank Diameter (m)	1								
	Initial Height (m)	1								
	Time Step (s)	10								
		Worksheet Implementation						VBA		Exact
	time (s)	h (m)	k1 (m)	k2 (m)	k3 (m)	k4 (m)		h (m)		h_{exact} (m)
	0	1	-0.11429	-0.11097	-0.11107	-0.10775		1		1
	10	0.888977	-0.10776	-0.10444	-0.10454	-0.10122		0.888977		0.888977
	20	0.784485	-0.10123	-0.09791	-0.09802	-0.09469		0.784485		0.784485
	30	0.686524	-0.0947	-0.09137	-0.09149	-0.08816		0.686524		0.686524
	40	0.596094	-0.08816	-0.08484	-0.08496	-0.08163		0.595094		0.595094
	50	0.510195	-0.08163	-0.0783	-0.07844	-0.0751		0.510195		0.510195
	60	0.431827	-0.0751	-0.07176	-0.07191	-0.06856		0.431827		0.431827
	70	0.35999	-0.06857	-0.06522	-0.06539	-0.06203		0.35999		0.359989
	80	0.294683	-0.06204	-0.05868	-0.05887	-0.0555		0.294683		0.294683
	90	0.235908	-0.05551	-0.05214	-0.05235	-0.04896		0.235908		0.235907
	100	0.183664	-0.04898	-0.0456	-0.04584	-0.04243		0.183664		0.183663
	110	0.13795	-0.04245	-0.03905	-0.03933	-0.03589		0.13795		0.137949
	120	0.098768	-0.03592	-0.03249	-0.03283	-0.02935		0.098768		0.098766
	130	0.066117	-0.02939	-0.02592	-0.02635	-0.02279		0.066117		0.066114
	140	0.039998	-0.02286	-0.01932	-0.01991	-0.0162		0.039998		0.039993
	150	0.020414	-0.01633	-0.01265	-0.01357	-0.00946		0.020414		0.020403
	160	0.007378	-0.00982	-0.00568	-0.0077	#NUM!		0.007378		0.007343

b).

VBA code called from worksheet cell formula.

```
Option Explicit

Function RungeKutta(t As Double, y As Double, dt As Double) As Double
'This function integrates a single ODE using a 4 term Runge-Kutta method.
'The ODE is defined in the private function dydt.

    Dim k1 As Double, k2 As Double, k3 As Double, k4 As Double

    k1 = dt * dydt(t, y)
    k2 = dt * dydt(t + 0.5 * dt, y + 0.5 * k1)
    k3 = dt * dydt(t + 0.5 * dt, y + 0.5 * k2)
    k4 = dt * dydt(t + dt, y + k3)

    RungeKutta = y + (k1 + 2 * k2 + 2 * k3 + k4) / 6

End Function

Private Function dydt(t As Double, y As Double) As Double

    Dim dhole As Double, dtank As Double
    Const g As Double = 9.8066

    dhole = 0.0508
    dtank = 1

    dydt = -(dhole / dtank) ^ 2 * Sqr(2 * g * y)

End Function
```

Figure 2. *Runge-Kutta solution of an ODE describing draining of a tank using a spreadsheet (a) and a VBA program (b).*

REFERENCES

1. Edgar, T., "Computing Through the Curriculum: An Integrated Approach for Chemical Engineering," CACHE Fall 2003 Newsletter, <http://www.che.utexas.edu/cache/newsletters/fall2003_cover.html>
2. Bloch, S.C., *Excel for Engineers and Scientists,* 2nd ed., Wiley, New York, (2003)
3. Filby, G., *Spreadsheets in Science and Engineering*, Springer-Verlag, New York, (1998)
4. Kral, I.H., *The Excel Spreadsheet for Engineers and Scientists*, Pearson Education, New Jersey, (1997)
5. Orvis, W. J., *Excel for Scientists and Engineers*, 2nd ed., Sybex, San Francisco, (1996)
6. Peress, J., "Working with Non-Ideal Gases," *Chem. Eng. Prog*., p. 39, March (2003)
7. Jevric, J., and M.E. Fayed, "Shortcut Distillation Calculations via Spreadsheets," *Chem. Eng. Prog*., p.60, December (2002)
8. Anthony, J., "Elements of Calculation Style," *Chem. Eng. Prog*., p.50, November 2001.
9. Chemeng Software, <http://www.chemengsoftware.com>
10. ChemSheet Software, <http://gttserv.lth.rwth-aachen.de/~cg/Software/ChemSheet/IndexFrame.htm>
11. Excel Unit Conversion, <http://www.unit-conversion.com/excel.htm>
12. The Chemical Engineers' Resource Page, <http://www.cheresources.com>
13. Beyond Technology, <http://www.beyondtechnology.com>
14. Emagenit, <http://www.emagenit.com>
15. Spreadsheet World, <http://www.spreadsheetworld.com>
16. Sauer, S.G., "Freshman Design in Chemical Engineering," *Chem. End. Ed*., **38**, 222(2004) ❐