# From C++ to Mathcad: Teaching an Introductory Programming Course with a Non-Traditional Programming Language

**K. P. Brannan, J. A. Murden**
**The Citadel**

## Abstract

Mathcad has replaced C++ as the language of the introductory programming course taught in the Civil and Environmental Engineering Department at The Citadel. Advantages and disadvantages associated with the switch are discussed in the paper. A comparison is made between the Mathcad-based programming course and the previous versions of the course taught using traditional programming languages. An evaluation of students' use of Mathcad a year after completing the Mathcad-based course is included and compared to those students who had taken the C++ version of the course. In addition, the classroom approach taken to teach the course is discussed.

## Introduction

Over the past few decades, engineering schools have placed a high priority on incorporating computer technology into the engineering curriculum. This has been no incidental achievement. The invention of the computer and the impact that the computer has had on numerical, informational, and graphical methods in teaching and research were included by the ASEE Centennial Recognition Committee in the ten most outstanding engineering education and engineering technology achievements of the past century[1]. Many schools began teaching computer skills early to ensure that students have a thorough exposure to the potential of the computer in engineering applications.

Introductory computer courses have historically involved the use of traditional computer languages such as FORTRAN, BASIC, Pascal, or C++ to solve engineering problems. Although FORTRAN has been in use for over forty years, it has undergone many changes and is still the language of choice of many engineering departments. As the use of the personal computer has grown, alternatives to FORTRAN such as BASIC and eventually Visual BASIC have gained acceptance. Currently, C++ is often considered the "language of choice" by many in the computer industry.

The introductory programming course in the Department of Civil and Environmental Engineering at The Citadel has undergone a number of transitions in the last ten to fifteen years. After teaching FORTRAN and/or BASIC on a VAX system for a number of years, an emphasis on desktop computing led to the adoption of QuickBASIC in 1990 as the department's standard programming language. In 1994, the Department adopted C++ as its standard programming language after a detailed evaluation[2]. C++ provided more extensive multi-platform support than had been available with the previous languages and also supported Object Oriented Programming.

Considering the popularity of C++ in the computer industry and the power and versatility of the language, there was no expectation that another change would be made for some time to come. Moreover, because of strong beliefs in the value of programming to a student's academic development and education, the idea of using anything other than a traditional programming language in an introductory course would have been rejected a few years ago. However, in recent years, a number of microcomputer application packages have been developed that combine numerical, graphical, and programming features which provide the user with unprecedented power for solving a wide variety of engineering problems and producing excellent desktop publishing. In the Fall of 1996, the department selected one of these packages (Mathcad 6+ Professional) as the primary tool for teaching engineering problem-solving and fundamental programming concepts. This paper will describe the reasons for the change to Mathcad along with the advantages and disadvantages of using Mathcad in an introductory programming course as compared to a traditional computer programming language. In addition, the classroom approach to teaching the course, including active learning techniques and team-teaching, will be described. Finally, an evaluation of student use of the package during the year following the course will be provided.

## Goals Associated with Teaching a Programming Language

One of the potential benefits of learning a programming language is that it can help a student enhance skills for solving engineering problems. Specifically, development of a full complement of programming skills requires a student to be able to examine a problem, develop logical approaches to solving the problem, select a solution, and test the solution. In addition, by learning to program, a student acquires the skills to develop custom tools to address unique needs for solving problems. Finally, programming provides the power to move well beyond what could be done manually, such as analyzing large amounts of information in a more exhaustive manner. All of these skills, associated with learning to program, can be valuable to an engineer in any discipline. Regardless of whether a traditional or a non-traditional programming language is used in an introductory programming course, the language should provide these opportunities. Thus, before the change to Mathcad was made, the faculty scrutinized the package to ensure that these goals could be met. Although a few disadvantages were noted that are discussed in later sections, it was felt that Mathcad clearly met these fundamental goals.

## Using Mathcad versus a Traditional Language

The department's introductory programming course focuses on developing solutions to problems using language independent algorithms. This is accomplished by using programming constructs that are common to most traditional languages. The programming fundamentals included in the three most recent versions of the programming course are presented in Table 1. A comparison of the topics covered in the different versions of the course clearly shows that the capabilities provided to the students have been consistent for all of the languages used. Because more topics are covered in the Mathcad based course, the percentage of time allocated to strictly programming topics has been somewhat less. On the other hand, Mathcad requires much less time and effort to document the solution and to format the output than traditional programming languages. Prior to adopting the Mathcad based course, some faculty expressed concern that students may be at a disadvantage later if faced with a situation where a traditional programming language was needed. Examples were offered such as developing stand-alone design tools as

Table 1. **Introductory Programming Course Topics using Mathcad, C++ and BASIC**

| Mathcad | C++ | BASIC |
|---|---|---|
| Numerical variable types, units | data types | variable types |
| Problem organization and printing | input and output | input and output |
| Branching (if, otherwise) | branching (if, if-else) | branching (IF-THEN-ELSE) |
| Looping (while, for) and implied loops(range variables) | looping (while, do-while, for) | looping (WHILE-WEND, DO-LOOP, FOR-NEXT) |
| File input and output | sequential files | sequential files |
| Vectors and matrices (single and double-dimension arrays) | single and double dimension arrays | single and double dimension arrays |
| User-defined functions | functions | functions and subprograms |
| 2D graphing, 3D graphing (surface and contour plots) | graphics | graphics |
| Data analysis and presentation (statistical functions, scatter plots, curve fitting, and histograms) | | |
| Symbolic, numerical, and graphical solutions | | |

senior projects or software development to support graduate research efforts. Ultimately, it was decided that the advantages of using Mathcad in the language-independent approach outweighed any perceived disadvantages. If at a later time students need to move from the Mathcad environment to a traditional programming language environment, it is a straightforward transition. Further, there is no need to try to anticipate the language that the students may need to use in the future.

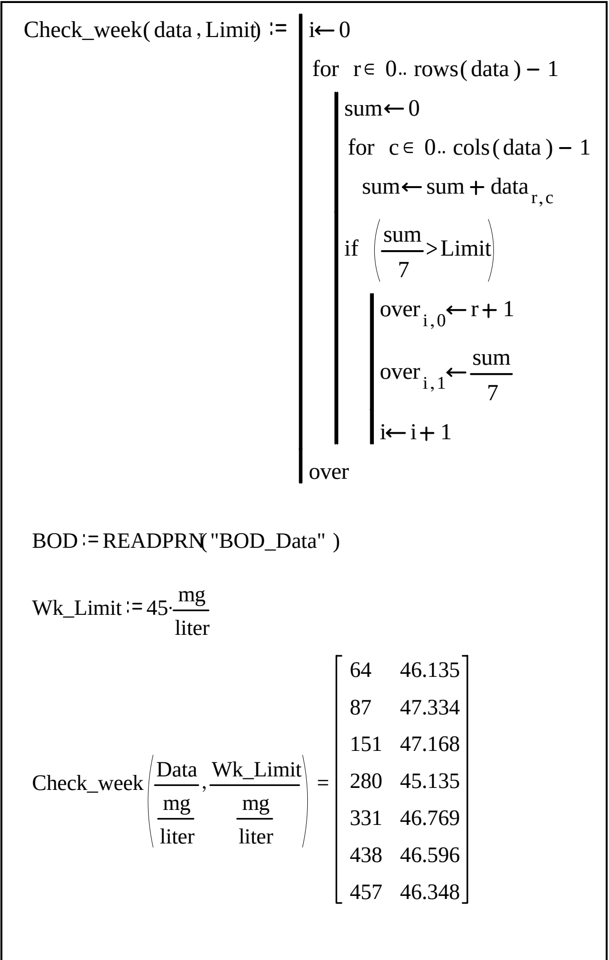
Extensive discussions have been held among the department's faculty members and also with engineering faculty from other institutions who are involved in educational software development. The software development manager of a large pharmaceutical firm who was conducting an Internet applications development workshop at the 1996 ASEE Southeastern Section Meeting made an interesting comment. After viewing some of the Mathcad solutions developed in the course, he was asked if they would be considered programming. He replied, "They look like the things we do in C or Java, so I'd call them programs."

The following is a portion of an assignment that was given during the Fall of 1997. The students were to develop a Mathcad solution using a multi-line function with for-loops and block-if statements that would read several years of daily readings from a disk file; then check to see if the weekly average of the readings exceeded some limit. For those weeks exceeding the limit, the weekly average and the week number were to be returned by the function. Mathcad statements from a typical solution to this portion of the assignment are shown in Figure 1.

Units may be conveniently incorporated, in general dimensional form, into Mathcad-based solutions. Students can take advantage of Mathcad's ability to monitor and convert units. When units are used, the dimensional consistency of the solution is ensured. By automatically converting units, Mathcad allows the values in a problem to be defined using the units that are most convenient; for example, lengths in feet, cross-sections in square inches, and loads in a user-defined unit, kips.

Symbolic manipulations can be performed in Mathcad. A strong argument can be made that the relationships in a system are best seen in a purely symbolic expression since all elements remain identifiable. One of the authors (Murden) emphasizes the value of symbolic solutions to students with the phrase "letters are your friends." With the availability of Mathcad to perform symbolic manipulations almost effortlessly, the students have accepted this phrase a bit more readily!

Graphing is well supported in Mathcad. Students routinely incorporate a graphical display of results into the problem solutions that have been developed using Mathcad. This can generally be accomplished with only one or two additional statements. Continuous functions and discrete data can be plotted conveniently. A wide range of graph types is available: x-y plots (linear or logarithmic scales), polar plots, surface plots, contours, 3D bar plots, 3D scatter plots and vector field plots.



```
Check_week(data, Limit) := | i ← 0
                           | for r ∈ 0.. rows(data) − 1
                           |   | sum ← 0
                           |   | for c ∈ 0.. cols(data) − 1
                           |   |   sum ← sum + $data_{r,c}$
                           |   | if $\left(\frac{sum}{7} > Limit\right)$
                           |   |   | $over_{i,0}$ ← r + 1
                           |   |   | $over_{i,1}$ ← $\frac{sum}{7}$
                           |   |   | i ← i + 1
                           | over
```

BOD := READPRN("BOD_Data")

$$Wk\_Limit := 45\cdot\frac{mg}{liter}$$

$$Check\_week\left(\frac{Data}{\frac{mg}{liter}}, \frac{Wk\_Limit}{\frac{mg}{liter}}\right) = \begin{bmatrix} 64 & 46.135 \\ 87 & 47.334 \\ 151 & 47.168 \\ 280 & 45.135 \\ 331 & 46.769 \\ 438 & 46.596 \\ 457 & 46.348 \end{bmatrix}$$

**Figure 1. Mathcad program example.**

Unfortunately, some things were lost with the shift to Mathcad. The most obvious is that the students no longer learn a traditional programming language. Second, stand-alone applications can not be created; Mathcad must be available to execute the worksheets that have been prepared. Third, since the contents of a worksheet are visible, the details of a solution cannot be hidden.

## Additional Benefits Associated with Mathcad

Student response was overwhelmingly supportive of the switch to Mathcad. Mathcad is a tool that the students begin to apply almost immediately (while they are taking the Mathcad-based class) in other courses, even without direction from the professors. A survey was conducted of students enrolled in civil engineering courses. Twenty-nine of the forty-three students who took the Mathcad-based course in 1996 completed the survey. Thirty-one of the thirty-seven who took the C++ based course in 1995 completed the survey. No attempt was made to contact students who had left the civil and environmental engineering department. A partial summary of the survey responses is presented in Table 2.

**Table 2. Student responses on Mathcad utilization.**

| | | Those Taught Mathcad | | Those Taught C++ | |
|---|---|---|---|---|---|
| I use Mathcad | very often | 7 | 24% | 7 | 23% |
| | regularly | 12 | 41% | 8 | 26% |
| | occasionally | 8 | 28% | 7 | 23% |
| | only if specified | 2 | 7% | 1 | 3% |
| I do not use Mathcad | | 0 | 0% | 8 | 26% |
| Total | | 29 | 100% | 31 | 100% |
| I develop solutions using Mathcad | | 26 | 90% | 23 | 74% |
| I use solutions developed by others | | 25 | 86% | 20 | 65% |

With 90% of the respondents who took the Mathcad-based course indicating that one year later they still voluntarily use Mathcad to develop their own solutions for problems, the faculty are convinced that this change has greatly increased the effective utilization of computers within the department. Similarly, nearly three-fourths of the respondents who took the C++ based course indicated that they have learned enough about Mathcad to develop their own Mathcad solutions to problems. This is in stark contrast to the 13 % of the respondents (2 from the Mathcad group and 6 from the C++ group) that developed solutions using C++ or some other traditional programming language on their own initiative. The most common comment made by those students is “we should have been taught Mathcad instead of C++.” Significant proportions of both groups make use of Mathcad solutions made available by faculty members.

Students’ measure of the significance of some of Mathcad’s capabilities can be gleaned from their responses to additional questions in the survey. Among students who had taken the Mathcad-based course, graphing, units and creating user-defined functions were the top three features. The students who had taken the C++ course considered units, symbolic manipulation and creating user-defined functions to be the most significant. Caution should be used before drawing conclusions from these differences. The students from the C++ course have been through an additional year of courses, which almost certainly has altered their perspectives.

The Mathcad environment is excellent for developing, documenting and presenting solutions. The required “language” is already familiar to the student: mathematics. Just as importantly, the mathematics look like the mathematics that the students have seen in their math courses. A full range of mathematical operators, built-in functions, and programming constructs is provided. It is unlikely that an engineering student will need an operator that would be unavailable. User-defined functions can be defined to perform “special purpose” calculations. Text capabilities allow all portions of the solution to be documented. Greek characters are supported in mathematical and text expressions. The graphical interface provides an onscreen presentation that corresponds to the printed output. This allows students to create and arrange the various elements in their solutions until the desired format is achieved prior to printing.

## Classroom Approach

The class was taught as a two semester-credit hour course, including one lecture hour and two laboratory hours. A typical class was designed to meet for a one-hour lecture on Monday followed by two one-hour laboratory sessions on Wednesday and Friday. Active learning techniques were used throughout the course, in both lecture and laboratory settings in an attempt to improve learning compared to a "typical" course, improve the grades of weaker students, increase retention of course material in the classroom, improve self-confidence of students, encourage supportive interaction among students in the academic environment, and increase interaction between faculty and students. Most of these are results that many claim can be achieved through effective cooperative or active learning techniques.

One of the first active-learning “lectures” was on ethical and professional issues associated with computer utilization. Students were divided into groups and given an ethical situation involving computer software to analyze. Following most of the elements that have been identified with effective active-learning groups[3, 4, 5], each member of the group was provided with a specific task to promote positive interdependence and interaction among the members of the group. Individual accountability was encouraged by randomly calling on any member of any group to explain the group’s solution. Some of the programming concepts were also taught in this manner, with groups analyzing or writing program segments.

Many of the lectures were taught in the laboratory using files prepared by the authors. These files, called electronic workbooks, were located on the department web-server. The workbook files contained appropriate explanations and interactive examples that provided students with the opportunity to acquire hands-on experience with the examples as they were discussed. Since the workbook files were downloaded to a diskette, students could review the workbook files interactively at any time during the lecture, or while working in the laboratory or in their rooms. Because Mathcad has a text editor, students had the option of taking notes electronically on the worksheet. Many students chose to print the workbook files and place them in a notebook.

To promote student-teacher interaction, the authors voluntarily team-taught in all classes held in the laboratory. This not only allowed more individual attention and more questions to be answered in class, but it allowed much more variety in classroom presentation, with the two professors changing “roles” in the presentation of the material. The authors have done this for a number of years and the students have consistently voiced their appreciation in student evaluations. The obvious disadvantage is that this takes a considerable amount of time on the part of the professors, but is felt that the benefits have been worth the effort.

Students were tested on course material in three ways. They solved problems by hand on paper; they developed Mathcad files on the computer that were printed and then submitted; and the students took web-based tests. For the web-based tests the students downloaded forms from the department’s web server and then submitted the completed forms back to the server. Although not enough data have been collected to draw meaningful conclusions, test performance associated with the three testing methods appeared to be comparable. The web-based tests initially took longer to prepare, but this was at least partially offset by the speed at which grading could be accomplished. It should also be mentioned that it takes time to develop appropriate questions for this mode of testing. Nevertheless, the authors were encouraged by how well web-based testing worked for the material in some portions of the course and plan to explore that testing method more in the future. Student response to the web-based tests was generally positive. However this may have been partially because of the novelty of the experience.

## Concluding Comments

Although there were a few disadvantages associated with the switch from C++ to Mathcad, in the opinion of the authors those disadvantages were more than offset by the numerous advantages provided by Mathcad. Student enthusiasm for the package has been much higher than has been observed for any of the traditional programming languages taught by the authors. Almost all of the students surveyed one year after completing the Mathcad-based course were still voluntarily using Mathcad in their classes. Two thirds of those students claimed to use Mathcad regularly or very frequently. Moreover, a high percentage of the students who took the C++ version of the course have been motivated to learn Mathcad on their own. Among the students from the C++ course, Mathcad is used far more frequently than C++ to develop computer-based solutions for assignments. Finally, in spite of the striking differences between the Mathcad environment and the even the newest development environments for traditional programming languages, programming concepts and algorithm development have been very successfully taught in a Mathcad-based course.

## References

[1] Burnet, George and Greisch, Janet Rohler, "The Ten Outstanding Engineering Achievements." *Journal of Engineering Education*, (January 1994).

[2] Murden, J. A. and Brannan, P. K., "C++ as a First Programming Language for Civil Engineers", *Proceedings, American Society of Engineering Education Southeastern Section Conference*, Greensboro, NC (1994).

[3] Smith, K. A., "Cooperation in the College Classroom," notes prepared in cooperation with David and Roger Johnson, University of Minnesota (1995).

[4] Smith, K. A., "Cooperative Learning: Effective Teamwork for Engineering Classrooms," IEEE Education Society/ASEE Electrical Engineering Division Newsletter (April, 1995).

[5] Johnson, D. W., Johnson, R. T., and Smith, K. A., "Active Learning: Cooperation in the College Classroom." Interaction Book Company, Edina, MN (1991).

## Biographical Information

KENNETH P. BRANNAN is an associate professor of Civil and Environmental Engineering at The Citadel. He has been elected President of the Southeastern Section of ASEE for the 1998-1999 term. He earned B.C.E and M.S. degrees from Auburn University and the Ph.D. from Virginia Tech. Dr. Brannan is a registered professional engineer with interests in freshman engineering education, water supply and wastewater treatment systems.

JOHN ALDEN MURDEN is an associate professor of Civil and Environmental Engineering at The Citadel. He earned the Ph.D. from Clemson University while studying wind effects on structures. Dr. Murden's technical interests currently include the effective application of computers in engineering education, system modeling, and experimental methods in structural mechanics. He is actively involved in the Southeastern Section of ASEE.