Computational Paradigms in Undergraduate Mechanical Engineering Education

B. K. Hodge, W. G. Steele
Mississippi State University

Abstract

Undergraduate mechanical engineering (ME) programs in the United States were surveyed to determine the usage of programming languages (such as C or FORTRAN) versus the use of arithmetic systems (such as Matlab or Mathcad). A survey form was e-mailed to all ME programs. The survey form was used to determine the following: (1) programming courses required, (2) use of programming in ME curricula, (3) use of arithmetic systems in ME curricula, (4) junior-level analysis courses required, and (5) computer ownership requirements. Seventy-three responses, representing a good cross section of ME programs were received. The survey showed that about three- fourths required at least one course in structured programming but that only one third of the programs requiring a formal programming course used programming in two or more required courses. More than three-fourths of all programs used arithmetic systems such as Matlab or Mathcad, and about the same number required a junior-level analysis course. Thirteen of the seventy-three ME programs that responded to the survey required students to own computes.

Background

At some point in any engineering endeavor, calculations must be made and "numbers" generated. The manner of doing calculations in the engineering workplace and in engineering education has continuously evolved, especially since World War II. Prior to that time, engineering calculations were accomplished in a manual fashion using what were then state-of-the-art techniques. In the eighteenth century, addition, subtraction, multiplication, division, and log tables were laboriously used. The situation was greatly improved by the slide rule [the first useful one was introduced about 1850 (1)] and the mechanical calculator, but generating numbers was, until after World War II, still a labor-intensive undertaking. Feynman's (2) anecdotal account of neutron diffusion calculations at Los Alamos in the 1940's is a good example of the drudgery and tediousness of extended pre-computer calculations. The digital computer fundamentally altered the use of "manual" calculations and replaced it with machine-based computations. Initial efforts were hard-wired (literally) with patch boards, but by the early 1950's higher-level programming languages evolved. For engineering computations, FORTRAN became the dominant programming language. However, as these advances were taking place, both the engineering workplace and engineering education struggled to effectively utilize the promise of the "computer" and to define the relationship between the computer and engineering. Indeed, one could argue that these struggles are ongoing.

The situation is much clearer today than a decade ago, and much progress has been made in effective utilization and in the computer/engineering symbiosis. Hodge (3) identified four

Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition Copyright 2001, American Society for Engineering Education

computational modes in the long-term evolution of Energy Systems Design (ESD), a specific mechanical engineering course at Mississippi State University. The course started out as hand-held calculator based (1981-1984) but quickly evolved into a course requiring extensive FORTRAN or BASIC programming to solve problems. During the late 1980's and early 1990's, the assignments moved from programming based to software application/modification based. However, by the mid-1990's, arithmetic systems such as Mathcad, Matlab, and Mathematica had been developed to levels that were very congruent with problem formulation and offered significant computational and visualization enhancements. Hodge's experiences with integrating Mathcad as the arithmetic engine in the ESD course are described in Hodge and Taylor (4). The use of Mathcad permitted much more realistic problems to be assigned and allowed the students to focus on problem formulation and solution (engineering tasks), rather than on programming issues. To some extent these four different modes of accomplishing arithmetic parallel the evolution of computational paradigms in engineering and engineering education.

Certainly, two dominant influences in the engineering workplace and in engineering education in the latter part of the twentieth century were the ever-increasing capability of digital computers and the ever-increasing utility of software. Indeed, engineering education has struggled to integrate and accommodate the evolving hardware and software into curriculums. The pre-EC2000 ABET accreditation criteria for a number of years specifically addressed the integration of computers into engineering curricula. Until the advent of arithmetic systems, such integration meant devoting time and effort to structured programming in higher-level languages (FORTRAN for example). Now the use of arithmetic systems such as Mathcad represents a more fruitful path that seems to portend the future. Indeed, as Baker et al. (5) point out, "…the days of amateur programming are over." That is to say, except for highly-skilled engineering specialists with post-BS degree education, engineers are not likely to do much structured programming in their careers. This does not imply that engineers won't use computers, only that applications, not programming, will be the engineering tasks.

All of the above raises questions about what is happening in mechanical engineering education in terms of computational paradigms used in undergraduate programs. Based on discussions at the last few ASEE Annual Meetings and Exhibitions, the topic of programming versus arithmetic systems is of great interest to the ME educational community. One way to find out what is happening is to survey the ME programs in the United States. This paper reports the results of such a survey.

Survey

A survey form was developed and transmitted via e-mail to all ME programs in the United States. The e-mail contained a letter explaining the survey form and the survey form. Issues of interested include:

1. Programming courses (FORTRAN, C++, ….) required.
2. Indications of programming language usage in required ME courses.
3. Arithmetic systems (Matlab, Mathcad, EES, Mathematica, ….) used.
4. Indications of arithmetic software usage in required ME courses.

5. Junior-level ME engineering analysis course required.
6. Computer ownership requirements.

The survey form is reproduced in Figure 1. The survey contains nine questions. Question 1 asked if a programming course is required in the undergraduate ME curriculum. If the answer is "yes," then questions 2, 3, and 4 as well as questions 8 and 9 are answered. If the answer to question 1 in "no," then questions 5, 6, 7, 8, and 9 are answered. This survey format was selected to ensure a clear distinction between programming usage and arithmetic system usage. Even so, interpretation differences were found in some survey comments. Several institutions, for example, considered Matlab to be a programming language. In the survey, the original intent was that Matlab usage was not structured programming. In compiling the survey results this ambiguity was noted. Question 2 requested the programming language (or languages used) in the required course. Question 3 asked if two or more required courses used programming, and question 4 asked if two or more courses used arithmetic systems. Both questions 3 and 4 relate only to programs that require a programming language course.

Questions 5, 6, and 7 are answered only if a programming language course is not required. Question 5 requested the system used if a programming language was not required. Question 6 asked if arithmetic systems were used in two or more required courses in the curriculum, and question 7 enquired if a special course was used to make sure all students were proficient in the arithmetic system used.

Question 8 was included in order to determine if a sophomore/junior-level engineering analysis course was required in the ME curriculum of the institution. Such a course would likely be taught by an ME faculty member and would emphasize algorithm applications (either in a structured programming language or an arithmetic system) of ME-oriented problems. Question 9 asked if computer ownership was required by students in ME at the school.

Mr. Thomas Perry, Engineering Education Director of ASME International, graciously agreed to e-mail the letter and survey form to all ME department heads via his ASME e-mail list for ME department heads. Additionally, he volunteered to pass out a hard copy of the survey form at an ASME department heads meeting in November. A total of 73 responses were received. The respondents included a good cross section of ME programs that were geographically well distributed. Based on the number of responding institutions requesting the results, considerable interest is present for this subject.

Survey Results and Interpretation

The response for each question in the survey will be examined and interpreted.

Question 1: Are all undergraduate mechanical engineering students required to take a programming course?

YES ➡ 58

NO ➡ 15

This response was mildly surprising as only 20 percent of programs did not require a structured programming course. The authors thought that the percentage not requiring a programming language would be significantly higher. However, in light of responses for questions 2 and 3, a better interpretation would be to count the 7 programs that required only Matlab as the programming language as “no” for this question and to add 7 to questions 5, 6, and 7. If that is done then

YES ➔ 51

NO ➔ 22

The purpose of this question was to ascertain programs requiring structured languages (C and Fortran, for example) and not arithmetic systems such as Matlab. The programming potential for Matlab is acknowledged, but for the purposes of this survey Matlab will be considered an arithmetic system.

Question 2: What computer language is emphasized in the programming course that is required?

C, C++ ➔ 35
Fortran ➔ 20
Matlab ➔ 11
Visual Basic ➔ 7

The response to this question is interesting for several reasons: (1) the total number of responses is greater than 58 as several schools specify more than one programming language, (2) a number of schools (11) consider Matlab as a programming language, and (3) C or C++ is the most popular language. The number of responses involving Matlab impacts questions 1 and 3.

Question 3: Is writing code/debugging extensively done by most mechanical engineering undergraduates in two or more required junior/senior mechanical engineering courses?

YES ➔ 19

NO ➔ 38

This is one of the most interesting of the survey responses as it shows that although 58 schools require a programming language, only 19 extensively use programming languages in two or more required courses. The inference is that although a programming language is required, the real purpose for the 38 schools is the discipline and logic that a structured programming language requires. Further analysis of the survey responses shows that three of the 19 “yes” responses involve Matlab only and not a structured programming language. Hence, a more accurate result would be

YES ➔ 16

NO ➔ 35

Question 4: Are arithmetic systems such as MathCad, EES, or TK SOLVER extensively used by most mechanical engineering undergraduates in two or more required junior/senior mechanical engineering courses?

YES → 42

NO → 16

This response is the most interesting since of the 58 schools requiring a programming language, 42 make extensive use of arithmetic systems. Only 12 of the 58 schools in this category indicated extensive (in more than two required courses) use of both programming languages and arithmetic systems. If the 7 schools that use Matlab only are taken out of this question and placed in questions 5, 6, and 7 the results are

YES → 35

NO → 16

Question 5: If writing code/debugging is not extensively used, what do most mechanical engineering undergraduates use for computations?

This is the first question for ME programs not requiring a structured programming language. From question 1, 15 schools answered as not requiring a programming language. All 15 indicated an arithmetic system was used in computations (see question 6). The arithmetic system choices were as follows:

Matlab	→	12
Mathcad	→	8
EES	→	3

The total is greater than 15 since many schools use two different arithmetic systems. With the 7 ME programs that identified Matlab a programming language included, the results become

Matlab	→	19
Mathcad	→	8
EES	→	3

Question 6: Are arithmetic systems such as MathCad, EES or TK SOLVER extensively used by most mechanical engineering undergraduates in two or more required junior/senior mechanical engineering courses?

YES → 15

NO → 0

When the 7 school using Matlab from questions 1 and 2 are included, more appropriate numbers are

YES → 22

NO → 0

Question 7: Does the department require a course whose prime purpose is to introduce mechanical engineering undergraduates to the computational system used?

YES ➡ 10

NO ➡ 5

Two-thirds of the programs using arithmetic systems, have a required course whose prime purpose is to develop student proficiency in the system. If the 7 programs that indicated they were using Matlab only in questions 1 and 2 are considered, the results become

YES ➡ 17

NO ➡ 5

The logic here is that these 7 programs with a required course using Matlab only are likely teaching the course within the department (or college).

Question 8: Is a sophomore/junior-level mechanical engineering department "engineering analysis" course offered? Such a course would stress techniques used in the solution of realistic mechanical engineering problems and could be either programming (FORTRAN, C++) or arithmetic system (MathCad, EES,….) based.

YES ➡ 54

NO ➡ 19

Most of the 73 ME programs require an engineering analysis course.

Question 9: Are mechanical engineering students required to own a personal computer?

YES ➡ 13

NO ➡ 60

This response was another mild surprise. Less than 20 percent of the ME programs responding to the survey indicated that student ownership of computers was a requirement. The engineering education literature infers that student ownership would be more widespread. Of the 13 respondents requiring computers, the breakdown for table top vesus lap top is as follows:

Lap top	➡	3
Table top	➡	3
Table top or lap top	➡	7

Conclusions

The most surprising results of the survey were the number of ME programs requiring a structured programming course, the relatively few programs using structured programming in two or more courses, and the significant usage of arithmetic systems by the responding schools. FORTRAN has clearly been displaced, as least in undergraduate engineering education, from the once dominant position it occupied. Considering the number of different arithmetic systems and programming languages used by the responding schools, no one system or language has attained dominance. However, since arithmetic systems have been tenable for only a few years, the rapid penetration of arithmetic systems into ME curricula is really significant. A likely scenario is that arithmetic systems such as MathCad, Matlab, and EES will improve over time and continue to displace programming in mechanical engineering undergraduate programs. Many of these arithmetic systems are becoming highly integrated with word processor, plotting, symbolic manipulation, and computational capabilities. This level of integration will likely lead to more acceptance in engineering education as well as the engineering work place.

References

1. Kaplan, E., 1998, "Calculating Machines," WWW.webcom.com/calc/.
2. Feynman, R. P. and Leighton, R., 1991, *Surely You're Joking, Mr. Feynman!*, QPBC, Camp Hill, PA.
3. Hodge, B. K.,, 1998, "The Evolution of a Required Energy Systems Design Course," ASME Paper 98-WA/DE-9, presented at the 1998 IMECE, Anaheim, CA, November.
4. Hodge, B. K., and Taylor, R. P., 1998, "The Impact of MathCad in an Energy Systems Design Course," Proceedings of the 1998 ASEE Annual Conference and Exhibition, Session 2266, Seattle, WA, June.
5. Baker, A. J., Pionke, C. D., Taylor, M. J., and Luttrell, B., 1996, "FEM Brought to the Academic Engineering Desktop," Proceedings of the 1996 ASEE Southeastern Section Meeting, Gatlinburg, TN, April.

B. K. HODGE

B. K. Hodge is a Grisham Master Teacher, a Giles Distinguished Professor, a Hearin Professor of Engineering, and Professor of Mechanical Engineering at Mississippi State University (MSU). He received degrees in aerospace engineering from MSU (BS and MS) and in mechanical engineering from the University of Alabama (MS and Ph.D.) and has industrial experience with Thiokol and Sverdrup (AEDC). Since joining the MSU faculty in 1978 he has written two textbooks and has developed six new courses. Dr. Hodge has conducted research in a diverse range of thermal and fluid sciences subjects including enhanced heat transfer, thermal systems simulation, uncertainty in thermal systems design, and desiccant technology. He serves as Director of the MSU Industrial Assessment Center and was the ASEE Southeastern Section President for 1999-2000.

W. G. STEELE

W. G. Steele is a Giles Distinguished Professor and Professor and Head of Mechanical Engineering at Mississippi State University. He received degrees in mechanical engineering from MSU (BS) and North Carolina State University (MME and Ph.D.) and has industrial experience with Westinghouse (Bettis Atomic Power Lab). Since joining the faculty at MSU in 1979 he has pioneered the use of uncertainty analysis in experimentation and design, has authored the leading book on the uncertainty in experimentation, and has served on national and international committees concerned with uncertainty analysis. Dr. Steele teaches courses in the thermal sciences and experimental techniques and conducts related research. Prior to becoming department head, he served as undergraduate and graduate coordinator for the ME program at MSU.

Mechanical Engineering Education Computational Paradigm Survey

Name ____

School ____

Address ____

____

____

Telephone ______________________ **FAX** __________________________

e-mail _________________________

1. **Are all undergraduate mechanical engineering students required to take a programming course?**

   YES ________________ **GO TO Question 2**

   NO ________________ **GO TO Question 5**

ANSWER QUESTIONS 2-4 ONLY IF QUESTION 1 IS "YES."

2. **What computer language is emphasized in the programming course that is required?**

   ____

3. **Is writing code/debugging extensively done by most mechanical engineering undergraduates in two or more required junior/senior mechanical engineering courses?**

   YES __________________ **NO** __________________

4. **Are arithmetic systems such as MathCad, EES, or TK SOLVER extensively used by most mechanical engineering undergraduates in two or more required junior/senior mechanical engineering courses?**

   YES __________________ **NO** __________________

Figure 1. Survey form

ANSWER QUESTIONS 5-7 ONLY IF QUESTION 1 IS "NO."

5. If writing code/debugging is not extensively used, what do most mechanical engineering undergraduates use for computations?

6. Are arithmetic systems such as MathCad, EES or TK SOLVER extensively used by most mechanical engineering undergraduates in two or more required junior/senior mechanical engineering courses?

YES __________________ NO __________________

7. Does the department require a course whose prime purpose is to introduce mechanical engineering undergraduates to the computational system used?

YES __________________ NO __________________

TO BE ANSWERED BY ALL

8. Is a sophomore/junior-level mechanical engineering department "engineering analysis" course offered? Such a course would stress techniques used in the solution of realistic mechanical engineering problems and could be either programming (FORTRAN, C++) or arithmetic system (MathCad, EES,….) based.

YES __________________ NO __________________

9. Are mechanical engineering students required to own a personal computer?

YES ___________________ Desktop ______ or Laptop ______

NO ___________________

FAX: 662-325-7223 or e-mail: hodge@me.msstate.edu
steele@me.msstate.edu

Figure 1. Concluded