Women, Mathematics, and Computing

De Palma, Paul. (2006). Women, Mathematics, and Computing. In E. Trauth (ed.), *Encyclopedia of Gender and Information Technology*. Hershey, PA: Idea Group Reference.

INTRODUCTION[1]

In 1963, Betty Friedan wrote these gloomy words:

> The problem lay buried, unspoken, for many years in the minds of American women. … Each suburban wife struggled with it alone. As she made the beds, shopped for groceries, matched slipcover material, ate peanut butter sandwiches with her children, chauffeured Cub Scouts and Brownies, lay beside her husband at night--she was afraid to ask even of herself the silent question--"Is this all?"

The passage, of course, is from the *The Feminine Mystique* (Friedan, 1983: 15). Though, it took another decade for the discontent that Friedan described to solidify into a political movement, even in 1963 women were doing more than making peanut butter sandwiches. They also earned 41% of the bachelor's degrees. By 1995, the number of degrees conferred had nearly tripled. The fraction going to women more than kept pace at almost 55%. Put another way, women's share of bachelor's degrees increased by 25% since Betty Friedan first noticed the isolation of housewives. Consider two more sets of numbers. In 1965, 478 women graduated from medical school. These 478 women accounted for only 6.5% of the new physicians. Law was even less hospitable. Only 404 women, or just 3% of the total, received law degrees in 1965. By 1996, however, almost 39% of medical degrees and 43% of law degrees were going to women (Anderson, 1997).

If so many women are studying medicine and law, why are so few studying computer science? A good question, and one that has been getting a lot of attention. A search of an important index of computing literature, the *ACM Digital Portal* (ACM, 2005a), using the key words, "women" and "computer," produced 2,223 hits. Of the first 200, most are about the under representation of women in information technology. Judging by the volume of research, what

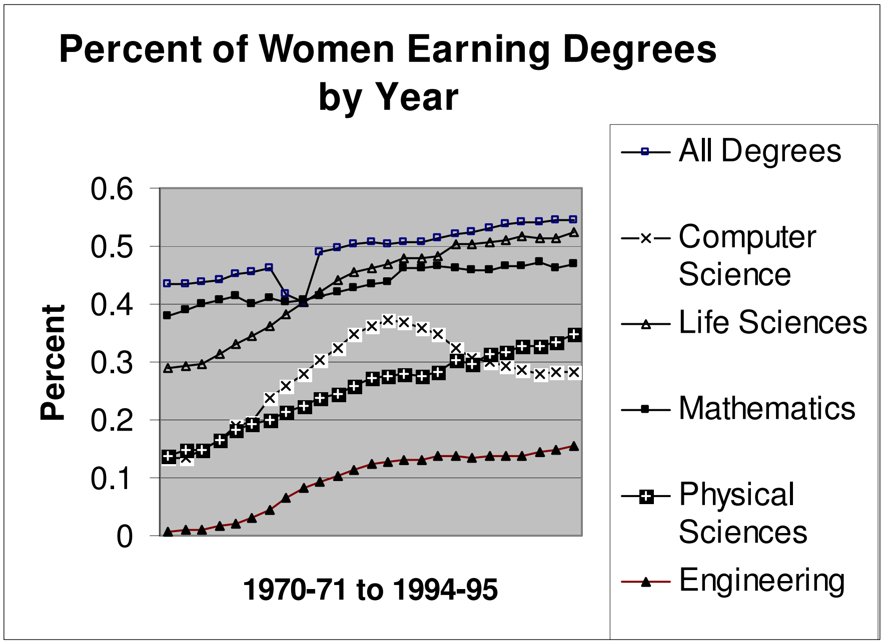
[1] This article grew out of a shorter opinion piece in the "Viewpoint" column of the *Communications of the ACM* (De Palma, 2001).

we can do to increase the numbers of women studying computer science remains an open question.

While most investigators fall on one side or the other of the essentialist/social constructivist divide (Trauth, Quesenberry, and Morgan, 2005), this paper sidesteps the issue altogether in favor of offering a testable hypothesis: girls and young women would be drawn to degree programs in computer science in greater numbers if the field were structured with the precision of mathematics. How we arrived at this hypothesis requires a look at the number of women earning degrees in computer science historically and in relation to other, apparently similar, fields.

BACKGROUND

In 1997, *The Communications of the ACM* published an article entitled "The Incredible Shrinking Pipeline" (Camp, 1997). The article points out that the fraction of computer science degrees going to women decreased from 1986 to 1994. This bucks the trend of women entering male-dominated professions in increasing numbers. The graph below shows the percent of women earning degrees in various scientific disciplines between 1970-71 and 1994-95 (National Center for Educational Statistics, 1997).



If you did not look at the data over time, you would be justified in concluding that the 13% or so of engineering degrees going to women represents a terrible social injustice. Yet the most striking feature of the degrees conferred in engineering and the physical and life sciences is how closely their curves match that of all degrees conferred to women. Stated another way, the fraction of degrees in engineering and the sciences going to women have increased enormously in a single generation. It has, in fact, out-paced the fraction of all degrees going to women. The curves for engineering and the life sciences both have that nice S shape that economists use to describe product acceptance. When a new kind of product comes to market, acceptance is initially slow. When the price comes down and the technology improves, it accelerates. Acceptance finally flattens out as the market becomes saturated. This appears to be exactly what has happened in engineering. Following the growth of the women's movement in the early seventies, women slowly began to account for a larger share of degrees conferred. By the early eighties, the fraction grew more rapidly and then, by the nineties the rate of growth began to slow. A parallel situation has occurred in the life sciences, but at a much higher fraction. Women now earn more than 50% of undergraduate degrees in biology.

Computer science is the anomaly. Rapid growth in the mid-eighties was followed by a sharp decline. The fraction of women graduating in computer science flattens out in the nineties. What's going on here? A study of German women noticed that the sharp increase in the number of degrees in computer science going to women followed the commercial introduction of the microcomputer in the early eighties (Oechtering, 1993). This is a crucial observation. In a very few years, computers went from something most people were only vaguely aware of, to a consumer product. What the graph does not tell you, is that great numbers of men also followed the allure of computing in the early and mid-eighties—numbers that declined by the end of the decade.

Despite many earnest attempts to explain why women don't find computer science as appealing as young men (see, for example, Bucciarelli,1997 and Wright, 1994), it is important to point out that computer science is not like the other areas we have been considering. Unlike physics, unlike chemistry, unlike mathematics, unlike electrical engineering, there is not an agreed-upon body of knowledge that defines the field. An important textbook in artificial intelligence, for instance, has grown three-fold in ten years. A common programming language used to teach introductory computing barely existed a decade ago. Noam Chomsky has suggested that the maturity of a scientific discipline is inversely proportional to the amount of material that forms its core. By this measure, computer science is far less mature than other scientific and engineering disciplines.

Many studies have shown that girls are consistently less confident about their abilities in mathematics and science than are boys, even when their test scores show them to be more able (see Mittelberg and Lev-Ari, 1999, for example.). Other studies attribute the shortage of women to lack of confidence along with the perception that computing is a male domain (Moorman and Johnson, 2003). Unfortunately, computer science, at least as presently constituted, requires a good bit of confidence. The kinds of problems presented to computer science majors tend to be open-ended. Unlike mathematics, the answers are not in the back of the book—even for introductory courses. There is often not a single best way to come up with a solution and, indeed, the solutions themselves, even for trivial problems, have a stunning complexity to them. The tools that students use to solve these problems tend to be vastly more complex than the problems themselves. The reason for this is that the tools were designed for industrial-scale software development. The move over the last decade to object-oriented languages has only exacerbated an existing problem (Hsia, Simpson, Smith and Cartwright, 2005). A typical lab assignment to write a program in the C++ or Java language will require that the student have a working knowledge of an operating system, a graphical user interface, a text editor, a debugger and the programming language itself.

One surrogate for complexity is the size of textbooks. Kernighan and Ritchie's classic, *The C Programming Language* (Kernighan and Ritchie, 1978) is 228 pages long. The first program in the book, the famous "Hello world," appears on page 6. Deitel, Deitel, Lipari, and Yaeger's (2004) *Visual C++ .NET: How to Program,* on the other hand, weighs in at a hefty four pounds and runs to 1,319 pages. Students have to wade through fifty-two pages before they reach the book's program equivalent to "Hello, world." The key to successful mastery in this environment is the willingness to tinker and the confidence to press forward with a set of tools that one only partially understands. Although we exhort our students to design a solution before they begin to enter it at the keyboard, in fact, the ready availability of computers has encouraged students to develop a trial and error attitude to their work. Those students willing to spend night after night at a computer screen acquire the kind of informal knowledge that is necessary to write successful programs. This is a world that will welcome only very self-assured young women.

MATHEMATICS, ENGINEERING, AND TINKERING

Recall Chomsky's observation that the most mature disciplines are the most tightly defined. What discipline can boast the tightness and precision of mathematics? As it happens, many reasonable people have attributed at least some of the shortage of women in science and computing as well as the less than positive attitudes toward computers to so-called math anxiety among girls. See, for example, Chang (2002), Jennings and Onwuegbuzie (2001), and Mark (1993). One study says that "The culture of engineering places particular stress on the importance of mathematical ability. Math is both the most complicated and the purest form of mental activity. It is also the most 'masculine' of subjects" (McIlwee and Robinson, 1992: 19, referring to Hacker, 1981). At first glance, the heavier reliance on mathematics might appear to explain why women avoid physics and electrical engineering while embracing biology and oceanography. But this explanation is insufficient for the simple reason that women receive nearly half of the undergraduate degrees in mathematics itself and were receiving almost 40% of them well before the women's movement became a mass phenomenon.

Here, then, is a hypothesis. What if the precision of mathematics is exactly what has appealed to women for so long? And what if the messiness of computing is what has put them off? So far, so good, but we still have to account for electrical engineering and physics. These have a smaller fraction of women than computer science, but are well-defined and rely heavily on sophisticated mathematics. What is it about physics and electrical engineering that women find unattractive? The answer is really quite simple. Students drawn to engineering and physics like to tinker with gadgets. See, for example Crawford, R., Wood, K., Fowler, M., Norell, J., (1994, p. 173.). That paper describes a grade school curriculum designed to encourage young engineers. It relieves heavily upon “levers, wheels, axles, cams, pulleys, forms of energy to create motion, etc.” McIlwee and Robinson (1992) report that 57% of male engineers surveyed chose the field because they like to tinker. Only 16% of women surveyed chose engineering for this reason. It should come as no surprise that the men associated with the microcomputer, the Bill Gates, the Paul Allens, the Jobs and the Wozniaks, all got their start as tinkerers. And as all parents know, but are hard-pressed to explain, their infant sons are drawn to trucks more readily than their infant daughters (Serbin, L., et al., 2001).

MICROCOMPUTERS, A PROBLEM WITH COMPUTER SCIENCE EDUCATION

Here we find a convergence with computer science, and, finally, an explanation for the steep rise in the number of women in the field following the introduction of the microcomputer and its drop a few years later. The development of the microcomputer changed computing enormously. In 1971, a small number of computer science departments awarded fewer than 2400 degrees. Most people who worked in the thriving data processing industry had received their training in the military, in for-profit vocational schools, or on the job. By 1986, that number had jumped to nearly 42,000, including almost 15,000 women. Clearly, the microcomputer played a large part in the growth of the academic discipline of computing. Like the dot-com boom, the growth could not be maintained. If the production of computer science degrees had continued to climb at the rate it climbed between 1975 and 1985, by 2001 every American would have had a B.S. in the field. In fact, the number of degrees awarded began to drop sharply in 1987.

We know why both men and women entered IT in the eighties. Why did the numbers drop by the late eighties? We can't really know the answer to this, of course, short of polling those who did not major in computer science during that period. But we can guess. Computer science is hard. What's more, it is not a real profession. There are no licensing barriers to entry, an issue that has been hotly debated in the computing literature for at least two decades (ACM, 2005b). Until computing societies agree on licensing and convince state legislatures to go along, students need not earn a degree in computer science to work in the field.

These things are equally true for women, of course, but the tinker factor is an additional burden. Before the mid-eighties and the mass availability of microcomputers, programmers could almost ignore hardware. The author wrote programs for a large manufacturer of mainframe computers in the early eighties without ever having seen the computer he was working on, nor, for that matter, the printer that produced the green bar paper delivered to his cubicle every two hours. There was tinkering going on in those days too, of course. But it was all software tinkering. Only computer operators touched the machine. The micro changed all that. Suddenly, those young men who had spent their adolescence installing exotic operating systems and swapping memory chips were in great demand. By adding hardware tinkering to the supposed repertoire of skills necessary to program, the microcomputer reinforced the male-dominated culture of IT (for an account of this very male atmosphere, see De Palma, 2005

CONCLUSION

Until the day when baby girls like gadgets as much as baby boys, let's look to mathematics itself to see what we can do about attracting young women to computer science. Well before other fields welcomed women, a significant fraction of degrees in mathematics were going to females. Let's assume that the mathematicians have been on the right track all along. A testable hypothesis presents itself. If we make computer science education more like mathematics education, we will make computer science more appealing to women. Computer science grew out of mathematics. How do we get back to basics?

First, teach girls who like to manipulate symbols how to program. Programming is weaving patterns with logic. If girls can do calculus, they can write programs. Second, try not to stray from logic. If we make computer science education less dependent on complex software tools, we remove some of the barriers between the student and logic. Third, minimize the use of microcomputers. Microcomputers, for all their cleverness, misrepresent computer science, the study of algorithms, as hardware tinkering. Fourth, ask students new to computer science to write many small functions, just as students of mathematics work countless short problems. Since there is something about the precision of mathematics that young women seem to like, let's try to make computing more precise. Later, as their confidence grows, they can take on larger projects. Fifth, regard programming languages as notation. It could well be that for complex systems, modern languages will produce a better product in a shorter time. But students don't produce complex systems. They produce relatively simple systems with extraordinarily complex tools. Choose a notation that is appropriate to the problem and don't introduce another until students become skilled programmers. Taken together, these suggestions outline a program to test the hypothesis.

Suppose we test the hypothesis and it turns out to have been correct. Suppose that, as a result, we give computing a makeover, and it comes out as clearly defined and as appropriate to the job as mathematics. Now imagine that able young women flock to the field. How might this change computing? To begin, students will no longer confuse half-formed ideas about proprietary products with computer science. Nor will they confuse the ability to plug in Ethernet cards with system design. It might mean that with a critical mass of women holding undergraduate degrees in computing, systems will be designed, not by tinkerers, but by women (and men) for whom the needs of computer users are front and center. Since stories of systems that failed through an over fondness for complexity are legion (De Palma, 2005), the makeover might even reduce the number of jerry-rigged systems. Thus does social justice converge with the market place—a very happy outcome, indeed.

REFERENCES

ACM (2005a). Search using key words: “women” and “computer.” *The Digital Library*. Retrieved 8/23/2005 from http://www.acm.org.

ACM (2005b). Search using key words: “license” and “profession.” *The Digital Library*. Retrieved 8/27/2005 from http://www.acm.org.

Anderson, Charles. (1998). *Fact Book on Higher Education: 1997 Edition.* American Council on Education. Phoenix: Oryx Press.

Bucciarelli, L. and Kuhn, S. (1997). Engineering education and engineering practice: improving the fit. In S.R. Barley & J. Orr (Eds.), *Between Craft and Science: Technical Work in U.S. settings* (pp. 210-229). Ithaca, NY: Cornell University Press.

Camp, Tracy (1997). The incredible shrinking pipeline. *Communications of the ACM,* 40 (10), 103-110.

Chang, J. (2002). Women and minorities in the sciences, mathematics, and engineering pipeline. *ERIC Clearinghouse for Community Colleges*. Retrieved 2/23/2006, from ERIC Digest (ED467855).

Crawford, R., Wood, K., Fowler, M., Norell, J. (1994, April). An engineering design curriculum for the elementary grades. *Journal of Engineering Education*, 83(2), 172-181.

Deitel, H., Deitel, P., Liperi, J. and Yaeger, C. (2003). *Visual C++ .NET: How to Program.* Upper Saddle River, NJ: Pearson Education, Inc.

De Palma, P. (2001). Viewpoint: Why Women Avoid Computer Science. *The Communications of the ACM*, 44 (6), 27-29.

De Palma, P. (2005). The Software Wars. *The American Scholar*, 74 (1), 69-83.

Friedan, Betty (1983). *The Feminine Mystique*. NY: W.W. Norton & Co.

Hacker, Sally (1981). The culture of engineering: women, workplace and machine. *Women’s Studies International Journal Quarterly*, 4, 341-53.

Hsia, J., Simpson, E., Smith, D., Cartwright, R (2005). Taming Java for the classroom. *ACM SIGCSE Bulletin , Proceedings of the 36th SIGCSE technical symposium on Computer science education,* 37 (1), 327-331.

Kernighan, B, Ritchie, D. (1978). The *C Programming Language*. Englewood Cliffs, NJ: Prentice-Hall.

Mark, J. (1993). Beyond equal access: gender equity in learning with computers. *The Eisenhower National Clearinghouse for Mathematics and Science Education.* Retrieved 8/27/05 from : http://www.enc.org/topics/equity/articles.

McIlwee, J. & Robinson, J. G. (1992). *Women in Engineering: Gender, Power and Workplace Culture*. Albany, NY: State University of New York Press.

Mittelberg D. & Lev-Ari L. (1999). Confidence in mathematics and its consequences: gender differences among Israeli Jewish and Arab youth. *Gender and Education*, 11 (1), 75 – 92.

Moorman, P., Johnson, E. (2003). Still a stranger here: attitudes among secondary school students towards computer science. *Proceedings of the 8th annual conference on innovation and technology in computer science education*, 193-97.

National Center for Educational Statistics. (1997). *Digest of Educational Statistics NCES 98-015*. Washington, D.C: U.S. Government Printing Office.

Oechtering, V. and Behnke, R. (1995). Situations and advancement measures in Germany. *Communications of the ACM*, 38 (1), 75-82.

Serbin, L., Poulin-Dubois, D., Colburne, K., Sen, M. Eichstedt, J. (2001). Gender stereotyping in infancy: visual preferences for and knowledge of gender-stereotyped toys in the second year. *International Journal of Behavioral Development*, 25 (1), 7–15.

Trauth, E., Quesenberry, J., Morgan, A. (2004). Gender issues in IT organizations. *Proceedings of the 2004 SIGMIS Conference on Computer Personnel Research*, 114-119. NY: ACM Press.

Wright, R., Jacobs, J. (1994). Male flight from computer work. *American Sociological Review*, 59, 511 - 536.

TERMS AND THEIR DEFINITIONS

Ethernet Card
Hardware that allows a computer to be attached to a network of computers.

Computer Science
An academic discipline that studies the design and implementation of algorithms. Algorithms are step-by-step procedures for solving well-defined problems. A precise description of a technique for putting words in alphabetical order is an algorithm.

Microcomputer
Also called a personal computer. The machine on your desk is a microcomputer.

Memory Chip
An informal term for RAM, or random access memory, or just plain memory. It is internal to a computer and loses its contents when the power is shut off. Programs must be loaded into RAM to execute.

Operating System
The collection of programs that controls all the computer's hardware and software. Important operating systems are Windows XP and Unix.

Program
A sequence of instructions that tells a computer how to accomplish a well-defined task.

Programming Language
The notational system that a programmer uses to construct a program. This program is transformed by another program, known as a compiler, into the instructions that a computer can execute. Important languages are Java and C++.