

# **General Report On The History And Status Of Computers Related To Mathematics**

**Prepared as partial completion of a  
sabbatical for the Fall 2003 Semester**

by

John Kennedy  
Mathematics Department  
Santa Monica College  
1900 Pico Blvd.  
Santa Monica, CA 90405

**`rkennedy@ix.netcom.com`**

## Introduction

During the Fall 2003 semester, John Kennedy in the Mathematics Department of Santa Monica College worked on a sabbatical whose purpose was to study and report on the topic known as MathML. MathML stands for Mathematical Markup Language and is a worldwide formal standard for encoding mathematics in an electronic form for the purpose of computer processing and communication, usually accomplished through using the Internet and its related technologies.

This brief document is written for the non-technical lay person who would like to read a brief history of early computers and contains a short summary of John's sabbatical work. There is another 117 page technical report that represents John's entire MathML findings, but that report is intended for math and computer science teachers and students interested in learning some of the technical details of the actual MathML language. By contrast, this report is deliberately brief, contains mostly history, and summarizes significant events and people from a mathematician's perspective.

## Brief Computer History Highlights

Most people would associate electronic computers and math because of their close historical ties. Historically, the first electronic general purpose computer in the United States was named the ENIAC and was built during World War II to calculate firing tables for artillery shells for the U.S. Army at the Aberdeen Proving Ground. Before this era, the word "computer" referred to people, not machines, who performed numerical computations. The impetus for this new form of electronic computing was to be able to quickly make the tables that might otherwise require years of human labor. In the midst of the war the Army had neither time nor labor to waste. However, mathematicians were involved with computing hundreds of years before the first electronic computers were invented. This report will briefly highlight some of the machines and people directly related to the development of the modern electronic computer.

Perhaps some might consider the first computer to be the sundial, others might like to think it was either pebbles or fingers, and still others might think it was the Chinese abacus. We will deliberately skip early history and consider the first "computer" to be a wooden mechanical machine made by the German Wilhelm Schickard in about 1623. This machine was apparently burned in a fire and the only evidence for its existence was discovered in the letters of the astronomer Johannes Kepler. This machine is notable because it supposedly could perform addition, subtraction, multiplication, and division. A replica has been made from the plans that were found in Kepler's letters.

The next machine that does survive to this day is a metal mechanical adding/subtracting machine built by the 19-year old French mathematician Blaise Pascal in 1642. Apparently there were many copies and variations of this machine that were made and sold commercially. Pascal is also the name of a modern computer language that was released in 1972. The Pascal name is in honor of the mathematician who lived 330 years earlier.

Thirty years after Pascal's first machine was made the German mathematician Gottfried Wilhelm Leibnitz invented a device known as the Leibnitz Wheel. This machine was better than Pascal's in that it could perform additions and subtractions fully automatically and it could also do multiplication and division. Leibnitz demonstrated this machine to the Royal Society in London in 1673. He also built a copy of this machine for Peter The Great to send to the emperor of China as a means of promoting

commerce between the West and the East. Leibnitz is credited along with Issac Newton as being one of the inventors of the branch of mathematics known as Calculus. Leibnitz was also known to be interested in reducing much of mathematics to simple logical operations with symbols. To this day we still use Leibnitz's system of notation in calculus.

## Charles Babbage

Skipping all the history of other early mechanical adding machines in the latter 1600's and the entire 1700's, we consider the next major developments to be the Difference Engine and the Analytical Engine designed by the Englishman Charles Babbage in the 1800's. The Difference Engine could calculate quadratic polynomials. Babbage was motivated by the need to make and print accurate navigation and other scientific tables. Books with tables of numbers had to be manually typeset and this process was prone to many human errors. Babbage knew that a machine that calculated and then printed tables would be more accurate. He actually designed a second difference engine that was never made but that could calculate polynomials up to degree six.

Babbage conceived the idea of his more powerful Analytical Engine as early as 1834. The Babbage Analytical Engine was to be controlled (programmed) to perform many different series of operations using punched cards. The idea is somewhat analogous to an old fashioned player-piano and is based on earlier punch card designs used in the Jacquard weaving loom in France. Although Babbage's Analytic Engine was never completed, it was the first calculating machine designed to incorporate the key idea of "programmed" control found in modern computers. Only small parts of this machine were made.

Countess Ada Lovelace, the daughter of Lord Byron, could be considered the world's first computer programmer because she actually translated an article describing how Babbage's Analytic Engine would work. She thus understood and promoted the concepts. The modern programming language named Ada is in her honor. If you ever visit the British Museum in London you can see models of both Babbage machines (constructed and completed after his death) along with a most wonderful collection of slide rules and many other examples of early special purpose computing devices.

Skipping even more mechanical calculator history and refinements, we jump to the German Konrad Zeus who built electromechanical computers in the late 1930's and early 1940's. Unfortunately his machines were destroyed by Allied bombing raids on Berlin in World War II, but some consider Zeus to be a possible co-inventor of the modern electronic computer. Another significant person was the American John Atanasoff who taught math and physics at Iowa State College and who along with his graduate assistant Clifford Berry created an electronic computer now known as the Atanasoff Berry machine. This machine was conceived in 1937 and was constructed by 1940. However, like Babbage's machines, this machine was never quite fully functional and it was limited to solving systems of equations. So although it could perform computations, it was not a general purpose machine as we think of computers today. Nonetheless, some consider Atanasoff as the forgotten father of the modern computer.

Skipping a little more history we come to Howard Aiken who taught applied mathematics in the Graduate School of Engineering at Harvard University. With the aid of the IBM corporation, he helped design and build what became known as the Harvard Mark I calculator. This electromechanical machine became operational in May of 1944 and was controlled by a large paper tape. The machine was over 50 feet long and 8 feet high. It was used by the Navy's Bureau of Ordnance.

Associated with the Mark I was Grace M. Hopper who made fundamental contributions to software design. She later became a Commodore in the Navy and helped standardize the COBOL computer language. She created the field of high-level programming languages in addition to creating the first computer compiler at Remington in 1952. For the non computer-literate, a compiler is a program that transforms human readable instructions into binary instructions that a computer understands. In 1954 Grace Hopper developed a program that could calculate derivatives of functions from the branch of mathematics known as calculus. For all her work she earned the nickname "Amazing Grace". In 1969 she won a "Man of the Year" award and in 1985 she was promoted to rear admiral. She retired in 1986 at the age of 79. She was the very last of the WWII Waves to leave the Navy and had an amazing career in two fields dominated by men.

It is difficult if not impossible for us to appreciate how these early machines were viewed at the time. In fact, when both the telephone and the automobile were invented people questioned why anyone would want one. These early computers were extremely expensive, had little or no memory, they were based on mechanical or electromechanical mechanisms, and were used almost exclusively for performing numerical computations. No wonder nobody could think of either wanting one or using one. Today we would be hard pressed to consider how we would live without computers. Either history changes us, or we change with history.

The first non-mechanical machine was the ENIAC and it was the influence of this machine that makes it so important in the history of computing. It used 18,000 high speed vacuum tubes as opposed to mechanical relays and so could perform computations hundreds of times faster than the Harvard Mark I. Its speed and size and capability were reasons why the ENIAC became famous as the first electronic computer. The ENIAC could multiply and divide and compute square roots and was controlled by punched cards that it could read at a rate of over 100 cards per minute. Programming this machine consisted of using cables to re-wire circuits. It had no form of memory as we now think of memory. It held only twenty 10-digit numbers, and was a monster at 100 feet long, 10 feet high, and consumed 140 kilowatts of power.

## **John von Neumann**

The famous mathematician John von Neumann is credited by some as designing the basic scheme of modern computers. He was Hungarian, earned his doctorate from the University of Budapest, and was considered a genius before he got involved with computers. He was at Gottingen University in Germany when he was asked to become a mathematical consultant for the secret Manhattan Project in 1943. Although it is still a point of controversy, some credit von Neumann as the person who came up with the idea of the stored-program computer as a result of his work associated with the ENIAC. The idea of an internally stored program did not exist at the time the ENIAC first became operational in the spring of 1945. However, the ENIAC was continually expanded and continued in service until 1955.

While the ENIAC was being produced there were many people involved in designing its successor that was called the EDVAC. EDVAC stands for Electronic Discrete Variable Calculator. Von Neuman helped write a 101-page report in 1945 that was titled "*First Draft of a Report on the EDVAC*". Herman Goldstine worked directly with von Neuman on both the ENIAC and the EDVAC and in his book Goldstine calls this report the most important document ever written on computing. Von Neumann fundamentally understood that instructions for controlling machines were as essential as the numerical

data that they contained and he came up with the idea of storing the instructions in an electronic form. Those instructions could then be read and executed at the same electronic speeds that were performing the computations. This idea seems trivial to anyone who understands how a modern microprocessor works, but in 1945 von Neumann may have been the first person to express the importance of this idea.

## **Alan Turing**

Now that we have reached the origins of the modern electronic computer we can step back a few years to consider the influence of the British mathematician Alan Turing. In 1936 Turing wrote a seminal paper that made him into one of the first and greatest theoreticians of computer science. He designed on paper a theoretical machine that later could be described as the modern digital computer. He was educated at King's College in Cambridge England. In 1936 he became a research student at the Institute for Advanced Study at Princeton University in the United States where he studied under the famous logician Alonzo Church. In this year, Turing conceptualized a theoretical device now called a Turing Machine. We must emphasize that a Turing Machine is not an actual device that you build and can hold in your hands. A Turing Machine is a conceptual virtual machine.

More important was that Turing found some types of mathematical problems that could not only NOT be solved by his conceptual machine, these problems would not be solvable by any machine built at any time in the future, including digital computers that had not even been invented yet! Turing was able to do this because his theoretical machine could be used to model any other theoretical digital machine, including machines that had not even been built. In other words, Turing discovered some of the fundamental theoretical limitations of what computers would be capable of doing. Turing is considered to be instrumental in properly and precisely defining the notion of a computer algorithm.

Turing returned to England in 1938 and became involved with breaking the codes that were used in the German Enigma machine. This was a secret device used by the German military to create secretly coded messages for commanders in the field, including those commanding the German U-boat submarines. Turing worked in Bletchley Park in England and supervised the building of the first English computer known as Colossus. Colossus was used to break the cyphers of the German Enigma machine. To this day parts of the Colossus project remain classified, but some consider the ability of the English to be able to read the German cyphers as the principal reason the Allies actually won World War II. Thus Turing's work is considered most profound.

## **Two More Hardware Advances**

By the mid 1950's the computer had been established as an indispensable scientific instrument. The theoretical basis for how the hardware should work was well established and the basic designs stabilized as the technologies being used matured. In this brief summary there are two more significant technical advances we need to point out. The first was the invention of the solid-state transistor that eventually replaced vacuum tubes. The early transistors operated 20 times faster than electronic tubes and led to the second advance that was the invention and development of integrated circuits.

The transistor was significantly smaller, was simpler to make, and consumed much less power than tubes. All three factors were responsible for the development of small portable radios in the mid 1950's. The transistor also made it possible to develop computers with larger and faster memories. However, as

computer circuits became larger and more complex, the problem of wiring all the needed interconnections between components became a major obstacle to further development. As the number of parts increased exponentially, it became increasingly difficult to build more complex circuits with hundreds of thousands of solder connections. Labor costs were another limitation and the number of wired interconnections became known as the problem of *the tyranny of numbers*.

The first integrated circuit made its debut in 1959 and was the creation of Jack Kilby who worked for the Texas Instruments company. It took a few years for solid state electronics to develop. The first simple micro-logic circuits contained perhaps four transistors and half a dozen diodes and cost \$120 each. By 1963 they were putting about 30 components in a chip for about \$32. The space program in the United States helped motivate the need for cheaper and smaller and more power-efficient electronics.

Because of the nature of the chip, wires were replaced by photo-lithographic techniques using sizes smaller than the width of a human hair. Robert Noyce of Fairchild Semiconductor is credited with developing the ideas that solved the wiring interconnection problem and the tyranny of numbers. By 1975 equivalent chip prices dropped to \$1.27 and it became routine to build memory chips with 65,536 transistors. In 1985 the Intel 386 microprocessor had 275,000 transistors. In 1995 the Intel Pentium Pro held 5.5 million transistors. In 2000 the Intel Pentium 4 was released with 42 million transistors. Continued growth like this fed the computer revolution. Such explosive growth was unimaginable by the creators of the ENIAC when they began construction work in 1943 on the first electronic behemoth.

At this point we will stop discussing the hardware and consider the significance of software. In the mid 1950's those involved with computers became aware that these machines could be used for doing more than just number crunching. Computers could be used to simulate parts of mathematics that dealt with symbolic manipulations as opposed to numerical computations. We already mentioned Grace Hopper's program to "*calculate*" derivatives of mathematical functions from calculus. This was one of the earliest examples of using a computer in a new and creative way. Perhaps now that computers are used as word processors and to display digital photographs and to play digital music it may be difficult to think there was a time when such possibilities did not even exist because the only information held in computers was considered to be numbers in digital form.

## Donald Knuth

We will also need to skip a lot of software history to get to the most preeminent scholar of computer science of the 20th century. This is Donald Knuth of Stanford University. Knuth was a mathematics major at Case Institute of Technology in Cleveland, Ohio. He so impressed his teachers that he was awarded a Master's Degree at the same time he earned his Bachelor's degree. In 1960 he began work on his Doctorate degree at the California Institute of Technology. He was awarded a doctorate degree in mathematics from Cal Tech in 1963. In 1968 Knuth went to Stanford University to teach Computer Science and the rest as they say is history. Knuth admired George Forsythe who moved from the UCLA Math Department to Stanford University in 1957. George Forsythe is credited with founding Computer Science as an academic discipline at Stanford in 1961. Before that time, even though computers existed, there were no departments of Computer Science, a fact that seems incredible today.

Knuth wrote the most significant work in the field of computer science in the 20th century. This is a series of books that is titled *The Art of Computer Programming*. In fact, Knuth's objective was to

summarize and categorize all that was known and had been discovered in computer science. His work is a massive one that is not even complete in the year 2004. He began writing in 1962. He originally outlined 12 chapters, but each chapter is sufficiently long so as to encompass more material than can be covered in even a one-year college course. Thus his work had to be broken down into a series of seven volumes. The first volume is only two chapters long, and if a student really studies it and works most of the problems it could take a couple of years to fully digest its contents.

In the middle of doing this massive work, Knuth spent 10 years developing systems for computer typesetting. His systems are extremely robust as he intended them to last 100 years into the future. Even more amazing is that Knuth did this work before the invention of modern display technologies and before the invention of the laser printer. His texts have been translated into multiple languages and still sell thousands of copies a month. Knuth has been awarded the National Medal of Science and in 1974 won the Turing Award which is the most prestigious award in the field of Computer Science. His influence in the field of analyzing algorithms is pervasive.

## **Of Calculators and PDA's**

As a math educator I have personally experienced the influence and development of calculators and personal computers. Although I learned to use a slide rule in high school, I was not that interested in performing numerical computations until I got my first little electronic calculator while I was in graduate school at UCLA. That first calculator cost nearly \$100 in 1973 and it would only add, subtract, multiply, and divide. Today you can buy a more powerful solar-powered model with hundreds of more functions for \$8 or less at any Sav-On Drug store. Using that first simple calculator I found that with some effort I could calculate trigonometric functions and logarithms.

In 1975 I bought my first programmable Hewlett-Packard calculator and became interested in learning how to program some numerical algorithms. This activity was just a few years before the personal computer explosion. However, this activity was responsible for stimulating my interests in programming and in learning more math. I also remember when math teachers were concerned that students were making too much use of calculators on homework and were also concerned about students using simple calculators on exams.

Now things have changed so much that high school students are encouraged to use not just ordinary scientific calculators, but graphing calculators. This is an issue for many math teachers because graphing functions without a computer is a skill we still teach in our pre-calculus and calculus courses. It is interesting to see the progression in the ten-year period between 1980-1990 from simple calculators to scientific calculators to graphing calculators to PDA's that contain a significant amount of software.

I think I can safely predict that PDA's (personal digital assistants) will be the end of scientific graphing calculators. All the smarts that are in graphing calculators are in the software and it is the software that will drive the future of small handheld devices. Why would anyone buy an expensive graphing calculator when for just a little more money they can just as easily get a combination of a PDA, a cell-phone, a GPS device, an AM-FM radio, a TV, an MP-3 player, and a photo and video display device, and a small but truly personal computer in the same size package?

## Math Computer Software Systems

This brings us to a new historical point where there will be a convergence of several technologies. This hasn't happened yet, but we are slowly progressing where all kinds of devices from refrigerators to toilets are controlled by small computers and are attached to small wireless networks. The personal computer and the software available for it have been growing at a rather rapid pace. Since most readers of this document are probably not familiar with professional math software, we should provide at least a brief background in that area.

In 1992 I applied for and was accepted in a special National Science Foundation program for community college math instructors. This program was held during the summer at my alma mater UC Santa Barbara. The purpose of this summer course was to introduce community college math instructors to the best available math computer software. Then, as well as now, the top three professional math software programs are named *Mathematica*, *Maple*, and *MatLab*. Then, if not more so now, these three programs are extremely powerful in terms of their mathematical capabilities. These are tools designed more for professional mathematicians to use in their work rather than for two-year community college students to use to learn math. Even with site licenses and student versions, the above three programs are also rather expensive for either students, or teachers, or schools to purchase and use on a continual basis. For example, the retail price of the full version of *Mathematica* is \$1,895.

We are going to single out the program named *Mathematica* and briefly discuss its significance in the field of mathematics. The first version of this program was released in 1988. The latest version is number 5 and after 15 years of development this has to be the most sophisticated mathematical software ever developed. There are three aspects of *Mathematica* that make it so exciting for math teachers and students to use. *Mathematica* is excellent at performing high speed and extremely large numerical computations, it is super excellent at doing symbolic manipulations that include abstract algebra, and it is excellent at making all kinds of 2- and 3-dimensional graphs.

The lay person can probably only imagine what we mean by symbolic manipulations and abstract algebra. To math people this means it goes beyond doing algebra and trigonometry as well as calculus. It covers these subjects by learning all the rules and being able to extend those rules. We can't give actual examples or even try to explain its full capabilities, but even highly trained mathematicians are truly amazed by *Mathematica's* capabilities. This is one area where software may change our conceptions of what mathematics is and what humans can do with it. It is also the area that comes closest to being able to perform mathematical thinking. We will have to leave the description at that.

### *Mathematica's* Founder

The *Mathematica* program was conceived and implemented by Stephen Wolfram. Wolfram was educated at Eton, Oxford and Caltech. He was 20 years old when he received his PhD degree in theoretical physics from Caltech. His interests have included cellular automata, complexity theories, and artificial life. He has taught at Caltech and the Institute for Advanced Study at Princeton and is a professor of Physics, Mathematics, and Computer Science at the University of Illinois. He has made major new discoveries that are described in his book *A New Kind of Science*. He is president and CEO of Wolfram Research, the company that produces the *Mathematica* program.

The *Mathematica* program has had a profound influence in the way computers are used in technical fields that are beyond mathematics. As a program, *Mathematica* is one of the largest single application programs ever developed. It is a major accomplishment in terms of software engineering. It is also supported by almost a hundred books and several journals, and each year there are world-wide conferences devoted to the system. By now you should have the idea that *Mathematica* is quite an accomplishment.

## The Internet

Nothing we have discussed so far has even touched on the Internet. There are perhaps only a half-dozen commercial programs that perform symbolic math. These programs are of course not compatible with one another. For this and other reasons, a natural need developed for mathematicians to communicate mathematical ideas using the Internet. How would this be possible with so many incompatible systems? And just how much math could be communicated anyway?

As a subject, math differs from history and biology and art and psychology because it often makes use of a complicated and sophisticated 2-dimensional writing system of notation that has been developed over centuries. Without knowing the meaning of the following equation, the question arises as to how you would write this equation using an ordinary word processor.

$$\mu = \frac{xy + \sqrt{9 - z^2}}{\sinh(x) \div \cos(y) + \ln\left(\frac{2 - \tan(x)}{\sqrt[3]{e^x + |x|}}\right)}$$

The short and simple answer is that you can't! While some may think you could just do a screen capture to make a picture of this equation, that would not be a good idea if you wanted to place this equation on an Internet web page. Why not? Because picture images cannot be searched for text, nor can they be sorted as text can be sorted, nor can they be indexed in a database without adding information.

## MathML At Last

In order to solve the problem as to how you could place the above equation on a web page a group of people got together to create a standard for representing mathematics on the Internet. The result was a new language that is named MathML. If you have used the Internet at all you might be aware that when you visit a web site your computer usually reads a special kind of file called an HTML file. HTML stands for Hypertext Markup Language. Unfortunately for mathematicians, HTML cannot format or use the above mathematical example equation. In fact, about the only support HTML has for displaying math is to do subscripts and superscripts. Of course the above equation was chosen because it goes beyond using just simple superscripts.

In the Internet world, the HTML language is still used, but it is being replaced by a better and newer technology that is called XML. XML is a somewhat strange language because it is what is called a meta-language. This means XML is a language that can be used to describe other languages. In particular, the variant of XML that can display the above equation is named MathML. You can just try to think that MathML is a particular kind or application of XML and XML is a newer and better HTML.

## My Sabbatical Results

My sabbatical proposal was to study and learn and report on the technical details of the MathML language. MathML along with other XML technologies make it possible to not only display complicated math expressions using native notations, it also makes it possible to search and sort and index mathematical information so that it can be more easily accessed and used. Beyond that, MathML also makes it possible for computers to actually understand the mathematical meaning of symbolic expressions. This is analogous to having a computer understand the following music score so that a computer could actually read the sheet music and then subsequently play the music it represents.



In a similar fashion, MathML is used by computers to read equations like the one shown above so that the computer can perform computations with it and make graphs. While the technical details of the MathML language cannot be explained in this short report, we can state that we found the MathML specification to be quite robust and interesting and useful.

The one negative that we found was that in order to make a true cut and paste system it requires a lot of other software to support the MathML standard. This is the one area where web browsers like Microsoft's Internet Explorer and the Netscape Navigator differ in how they handle the XML language. So while MathML is a wonderful standard and is supported by systems like *Mathematica*, it may take a while longer before the use of MathML becomes more widespread. Almost all commercial math programs are committed to supporting the MathML standard in the future.

## About The References

If you would like to read more about the history of computers we have provided a list of 20 references. We hope that you will take the time to browse a few of these books to learn more about the people and the nature of computing and its relationship to the field of mathematics. You can also consult my homepage web site to learn even more and download other papers and other free math software. You can also read the main report from my sabbatical. See also [homepage.smc.edu/kennedy\\_john](http://homepage.smc.edu/kennedy_john).

The books listed below are ordered by the author's last name. Most of these books are written for the lay person and are related in some way to the history of computing. The texts by Berlinski, Cajori, Dewdney, and Wolfram are exceptions to this statement in that they are much more technical and mathematical. This list does not represent a complete history, but each book is interesting, and as a whole these books should give you a different and more broad perspective on the nature and history of computing.

I especially like the text by Paul Ceruzzi for its perspective after 1945. Ceruzzi is a professional historian and is associated with the Smithsonian Institution's National Air and Space Museum. The Williams book covers machines from the beginning of time up to 1968, but it is especially useful for history prior to 1945. Goldstine's book is also particularly excellent for the period from 1944-1957 because he was personally involved with the most significant people and developments in that era. You will also find the book by Robert Slater more interesting if you prefer reading more about people as opposed to machines. The book by Dewdney is more interesting if you are interested less in people and machines and are more interested in software and the techniques and problems that are in computer science. The book by Reid is interesting because it focuses on the development of the computer chip which comes very late in the history of computing. Reid's book also contains an excellent list of references for later computer technology.

Technically, calculating devices developed from those that were purely mechanical with relatively few parts, to those that were electromechanical with relays and hundreds of parts, to those with vacuum tubes with thousands of parts, to those with transistors with hundreds of thousands of parts, and finally to those with integrated circuits, also called computer chips that helped solve the wiring interconnection problem, and that contain multi-millions of transistors.

## References

- [1] *The Advent of the Algorithm*, by David Berlinski, Harcourt, 2000. 345 pages.  
ISBN 0-15-100338-6
- [2] *When Computers Went To Sea*, by David L. Boslaugh, IEEE Computer Society Press, 1999.  
467 pages.  
ISBN 0-7695-0024-2
- [3] *A History of Mathematical Notations*, by Florian Cajori, Open Court Publishing Co., 1928, republished by Dover Publications, New York, 1993. 820 pages.  
ISBN 0-486-67766-4.
- [4] *A History of Modern Computing*, by Paul Ceruzzi, The MIT Press, 1998. 398 pages.  
ISBN 0-262-03255-4.
- [5] *Before the Computer (1865-1956)*, by James Cortada, Princeton University Press, 2000, 350 pages.  
ISBN 0-691-05045-7
- [6] *The Universal Computer*, by Martin Davis, W.W. Norton & Company, 2000. 257 pages.  
ISBN 0-393-04785-7
- [7] *The New Turing Omnibus*, by A.K. Dewdney, Computer Science Press, 1993. 455 pages.  
ISBN 0-7167-8271-5
- [8] *The Abacus*, by Jesse Dilson, St. Martins Press, 1968. 143 pages.  
ISBN 0-312-10409-X

- [9] ***The Computer: From Pascal to von Neumann***, by Herman H. Goldstine, Princeton University Press, 1980. 365 pages.  
ISBN 0-691-02367-0
- [10] ***Alan Turing The Enigma***, by Andrew Hodges, Simon & Schuster, 1983. 587 pages.  
ISBN 0-671-52809-2
- [11] ***Charles Babbage Pioneer of the Computer***, by Anthony Hyman, Princeton University Press, 1982. 287 pages.  
ISBN 0-691-02377-8
- [12] ***A History of Computing in the Twentieth Century***, edited by N. Metropolis and J. Howlett and Gian-Carlo Rota, Academic Press, 1985. 693 pages.  
ISBN 0-12-491650-3
- [13] ***Atanasoff, The Forgotten Father of the Computer***, by Clark Mollenhoff, Iowa State University Press, 1988. 274 pages.  
ISBN 0-8138-0032-3
- [14] ***The Chip***, by , T.R. Reid, Simon & Schuster, 1984. 243 pages.  
ISBN 0-671-45393-9
- [15] ***Out of Their Minds, The Lives and Discoveries of 15 Great Computer Scientists***, by Dennis Shasha and Cathy Lazere, Coernicus (Springer-Verlag), 1995. 291 pages.  
ISBN 0-387-97992-1
- [16] ***Engines of the Mind***, by Joel Shurkin, W.W. Norton & Company, 1996. 363 pages.  
ISBN 0-393-31471-5
- [17] ***Portraits In Silicon***, by Robert Slater, The MIT Press, 1992. 374 pages.  
ISBN 0-262-69131-0
- [18] ***History of Computing Technology***, by Michael R. Williams, IEEE Computer Society Press, 1997. 426 pages.  
ISBN 0-8186-7739-2
- [19] ***A New Kind of Science***, by Stephen Wolfram, Wolfram Media, 2002.  
1197 pages.  
ISBN 1-57955-008-8
- [20] ***The Mathematica Book (5th Edition)***, by Stephen Wolfram, Wolfram Media, 2003. 1488 pages.  
ISBN 1-579-55022-3