I'm not a robot

Digital technologies have transformed both the historical record and the historical profession. This Focus section examines how computational methods have influenced, and will influence, the history of science. The essays discuss the new types of questions and narratives that computational methods enable and the need for better data management in the history and philosophy of science (HPS) community. They showcase various methodological approaches, including textual and network analyses, and they place the computational turn in historiographical and societal context. Rather than surrendering to either technophilia or technophobia, the essays articulate both the benefits and the drawbacks of computational HPS. They agree that the future of the field depends on the successful integration of technological developments, social practices, and infrastructural support and that historians of science must learn to embrace collaboration both within and beyond disciplinary boundaries. Mechanical Aids to Computation and theDevelopment of Algorithms2. Early History2.1. IntroductionAlthough the computer and its widespread application in our society, arephenomena that have become predominant in only the last 30 years, many of theconcepts underlying these developments have their origins in concerns dating back to theearliest cultures. Computers manipulate data (Latin, plural of datum, neut. p.p. dare, cognateSanskrit, datta: those things which have been given), i.e. process and transform given representationsof information in order to obtain a desired result. Within this basic description of computerbehaviour we can discern two fundamental ideas:Representation: A concrete, symbolic encoding of information, e.g. numbers, words, names.Transformation: The steps (recipe, program, algorithm) used to calculate aspecific result.A symbolic encoding of information provides a vehicle for communication - informationcan be passed on in a commonly understood form. A record of the process by which the representationis transformed allows the calculation process to be carried out repeatedly on different sets ofdata, e.g. we have all learned the steps needed to determine the result of multiplying any two large numbers.One might ask, however, why, given a system for encoding information and the sequence of stepsneeded to manipulate this to a specific end, it should be necessary to seek mechanical assistancewith the task? The answer to this question lies in the fact that the calculations required to carryout these tasks are often laborious. This fact has two consequences if no mechanical aid is employed:The computation will take a long time to complete.The answers may be incorrect, because of human error, and so the same computation mayhave to be carried out several times.(Consider, which would you prefer: to multiply two 5 figure numbers by hand on paper or to use anelectronic calculator? Which answer would you have greater confidence in?)The calculations involved in predicting celestial phenomena from previously observed data; inassessing the rates of taxation to levy in order to raise a required sum; in analysing censusstatistics; in determining the path of a projectile; all of these are examples where lengthy,tedious and (if done by hand) error-prone computations arise.Thus the historical development of the topic we are considering can be seen to be based onthree related processes:The development of symbolic representations of information that are amenable tomanipulation.The formulation of algorithms by which such representations may be processed to solvecomputational problems.The construction of mechanical aids that allow such algorithms to be implemented in an efficientand a reliable manner.It ought to be clear that methods for representing information must been have developed first, so it isappropriate to examine one aspect of such representations - number systems - before going onto consider the algorithms and mechanical aids that utilise them.The simplest method of representing numbers is to use a sequence of identical marks todenote quantities, e.g. Figure 1 belowArchaeological discoveries have established that the `tally system' was independentlydeveloped by many early cultures, e.g. bones with notches denoting quantities have been foundby anthropologists in Czechoslovakia, similar fragments dating from around 8500 B.C. have also beendiscovered in Africa. Despite its rudimentary nature, this system does exhibit importantfeatures that were to be preserved in later systems. The most important of these is the conceptof counting in multiples of some basic number (5 in the example above). While most societiesadopted 5 or 10 as the typical base (from the practice of counting on fingers) these were by no meansuniversally chosen. The South American Mayan culture employed a system based around the number 360 (fromtheir estimate of the number of days in a year); the Babylonians used 60 as a base. It is interestingto note that the influence of the Mayan and Babylonian systems continues in the present day (a circle contains360 degrees, a degree 60 subparts called minutes, a minute 60 subparts called seconds; similarly we denotetime units in multiples of 60 - 1 hour is 60 minutes is 60 times 60 seconds)Although the tally system has several advantages - it is easy to understand, simple arithmeticoperations such as addition, subtraction and multiplication can be performed without greatdifficulty - it is extremely cumbersome when used to represent large numbers, such as might arisein recording population sizes, and it is not suitable for more complicated computational tasks, e.g.division. Attempts to address the first problem can be discerned in the notational systems usedby Greek and Roman societies (from ca. 1000 B.C for Greek, 700 B.C, Roman). These, though, were reallyonly slightly more sophisticated versions of the tally system: instead of using a single denotationalsymbol to construct numbers, a set of different symbols is employed, each representing a different quantity,Figure 2.1) {muupsilonrhoiotaoiota} is typically translated as {10,000}but is more accurately rendered as `countless'. Herodotus' estimate of the Persian forces atthe Battle of Marathon and the source of the English word `myriad'.Notice that both of these systems eventually become no better (and in some respectsconsiderably worse) than the simple tally system. The method of Roman numerals does, however,introduce an important idea, although failing to exploit it fully: the concept of positionalnotation. Thus in the representation of 1948 the opening M has a different meaningfrom the M occurring as the third character. Both of these systems indicate the importanceof the qualification `amenable to manipulation' that we stressed earlier. The major deficiencyof the Roman system is that it is decidedly unsuitable as a basis for computation. In the tally system the calculation of 494 + 506 is a (notationally) lengthy but very easy computation; the computation of fBCDXCIVfR + fBDVIfR in Roman numerals is not (the answer, M,bears no symbolic relation to the summands and although the result is almost twice as largeas the individual contributions, it is expressed using one symbol instead of nine). A task such as multiplication, conceivable in the tallysystem, present major difficulties in Roman numerals. So inflexible is this system thata suitable topic for doctoral research in the early European universities concerned algorithms formultiplication and division using Roman numerals. Despite all of these drawbacks, Roman numeralsremained the predominant means of representing quantities in European culture well into the 14th century.They were ultimately replaced by a system which contributed what was one of the most importantdiscoveries of early science: a fully positional notation with a representation for the number zero.There is evidence that the Mayan civilisation employed a symbol for zero in their number system. Itsarrival in European science came via knowledge of Arabic mathematics. Arab scholars had themselveslearned of this system from Indian civilisation. The Indian discovery dates from, at the latest,200 B.C (the earliest recorded use, in a textbook, by Bakhshali). The word `zero' in English, itselforiginates from India (Sanskrit sunya - empty or blank - translated as zifr byArab writers, hence Latin zephirum and English zero and cipher). The Arabicsystem employed 10 different symbols representing the numbers 0,1,2,...,9 and formed thebasis of the decimal system that is used today. First popularised in Europe by Leonardo of Pisain his Liber Abaci (The Book of Computation) of 1228, despite attempts to suppress it during thereaction against Islamic scholarship, the obvious advantages of the system over Roman numerals eventuallyled to its being universally adopted.We now turn to the development of the earliest algorithms and mechanical computing devices.Figure 3 shows the structure of one of the earliest forms of primitive computing device that could beused for addition and subtraction. Units were marked out on two lengths of wood. Addition tables andthe result of subtractions could be constructed by lining up appropriate units. Other than such basicmechanisms, look-up tables were constructed. At Senkereh in what was Babylonia a clay tabletdating from between 2300-1600 B.C has been discovered which contains the squares of the first 24 numbers.This is now in the British Museum. Such tables obviate the need to recompute particular values and wereprobably constructed using stones as counters, e.g. Figure 4.The use of pebbles and arrangements of these as aids to computing was common to many cultures. Itsinfluence is apparent in many terms still used today, e.g square (via arrangementsin Figure 4), triangular numbers which represent the sums of the first n numbers; calculusand calculation both deriving from the Latin for pebble (hence the different usages of the wordcalculus in mathematical sciences and medicine); cheque and exchequer which derive from the mediaevalEnglish custom of calculating tax levies by piling stones on a board marked out in black and whitesquares (the word is combination of Latin ex meaning `from' and Norman French cheque for the checkedpattern). The use of counters as an aid to computation reaches its greatest level of sophistication and powerin a device which is still very extensively used today in the Far East: the abacus. Figure 5The abacus represents the state of a calculation by the position of beads strung into columns on wires.The earliest invention of the abacus is now credited to the Babylonians (the word abacus derives from aPhoenician word abak). By the sixth century B.C. they were widely used in Greek society:the historian Herodotus (ca. 484-424 B.C) mentions them in this period and left records of complexcalculations performed on them, e.g. the accrued interest on a loan of 766 talents, 1095 drachmaeand 5 obols over a period of 1464 days at a rate of one drachma per day for every 5 talents. Other incidencesof the use of abaci in late Greek culture are Eutocius of Ascalon's computation of (30133/4)^2;references in the surviving speeches of the orator Demosthenes (385-322 B.C); and the writings of theCynic philosopher Diogenes (d. ca. 320 B.C). It was in Oriental cultures, principally China and Japan, however, thatthe abacus reached its highest level of development. The Japanese contributed the concept of dividingthe abacus frame horizontally into two zones (the so-called Heaven zone with 2 beads and EarthZone with 5)^22) Although the modern Japanese abacus has reduced the heaven zone to a single bead.With this device calculations of great complexity could be performed at great speed. In 1946, in a meetingbetween the fastest mechanical calculator operator in the U.S. Army, one Private. T.N. Wood equippedwith a contemporary state-of-the-art calculator, was defeated in4 out 5 speed contests by Kiyoshi Matsuzaki, who used an abacus. This device is still widely used inbanking and financial calculations in Japan today.It should be noted that even such crude devices as the notched sticks of Figure 4, presume someunderstanding of the algorithmic process determining their use. It is likely that the firstemployers of simple computational aids were unaware that the reason why they produced the correctanswer was because they exploited simple algorithms relating the data operated upon to the resultsobtained. The Greek mathematician, Euclid (4th Century B.C) is the earliest known identifier of specific algorithms (although it is clear that the calculations, such as those documented by Herodotus on abaci - computethe interest due on amount x at a rate of y per z for each w - presuppose methods stating howto proceed). Euclid's work The Elements is of collection of results, mainly in Geometry, spreadover 7 books. It is important, for our purposes, for two reasons:It is the earliest known attempt to formalise the concept of algorithm, i.e. to separate `admissible'computational processes from `inadmissible' ones.Book VII, dedicated to properties of numbers, describes a number of important algorithms at least oneof which is still taught today.As regards the first point, Euclid addressed the issue of what algorithms could be said to bevalid if one was concerned with constructing various geometrical objects, e.g. squares with a certainarea; lines and angles with particular properties; regular polygons, etc. The Elements admitssuch an object if and only if one can show how to construct it in a finite number of steps usingonly a ruler, i.e. straight edge to draw lines, and a pair of compasses (to draw circles and arcs). See Figure 6. For example one mayeasily check that the algorithm below constructs a regular hexagon - a six sided figure in whichall sides have equal length and all internal angles are equal:Draw a circle C of radius r (uses compasses)Choose a point phi on C and draw a circle with centre phi and radius r. The new circlecuts C at two points alpha and beta.Draw a circle with centre alpha and radius r. This cuts C at phi and a new point gamma.Draw a circle with centre gamma and radius r. This cuts C at alpha and a new point delta.Draw a circle with centre delta and radius r. This cuts C at gamma and a new point epsilon.Draw a circle with centre epsilon. This cuts C at delta and beta.Use a straight line to connect phi (-> alpha (-> gamma (-> delta (-> epsilon (-> beta (-> phiThe resulting polygon is a regular hexagon.Ruler and Compass Construction of Regular HexagonStages of this process are illustrated in Figure 7.Although this may seem a very primitive class of algorithm, problems of great subtlety arise in it.In particular, the problem of squaring the circle (i.e. construct a square whose area is equal to thatof a given circle) was first raised by Euclid. This was not shown to be impossible using ruler andcompasses until the end of the 19th Century^3 - over 2300 years after Euclid's death.3) By Lindemann in 1882. Of the other important constructions left open by Euclid -trisection of a given angle and duplication of a given cube - the former was provedimpossible by Descartes (1637), the latter was known to be impossible by Arabic mathematicians.The second important contribution of The Elements is the algorithm to calculate the greatestcommon divisor of two numbers. Given two numbers - m and n - the greatest common divisorof m and n, denoted gcd(m,n), is the largest number that divides both m and n without leavingany remainder, e.g. gcd (12, 40) = 4, gcd (7, 15) = 1, etc. Euclid's Algorithm, as it is nowknown, is shown below:greatest common divisor of m and n 1. If m is smaller than n then swop the values of m and n. 2. Set m equal to the remainder of m divided by n. 3. If m is not equal to 0 then go back to step 1, with the new values of n and m. 4. Return n as the answer.Euclid's AlgorithmEuclid's work was not the only early investigation of algorithms. Another important Greek contributionconcerns a problem which is still very much of interest today - the generation of prime numbers.A prime number is a whole number, greater than 1, which is divisible without remainder by onlyitself and by 1. Methods of finding prime numbers have existed for over 2000 years and, since theonly known algorithms are computationally very demanding, finding large primes is a standard performancetest carried out on new powerful computer systems: in the last 40 years the size of the largest known primehas advanced from a 40 digit number (found in 1886) to a 15,000 digit number (identified in 1991).The first algorithm to identify prime numbers is the famous Sieve of Eratosthenes (fl. 3rd Century B.C).This works as follows: suppose on wishes to know all the prime numbers less than some number, n say.Write down all the numbers from 2 to n and then carry out the following steps:1. Let k := 22. For each number, m, between k+1 and n if m is an exact multiple of k then cross m of the list of numbers.3. Set k to be the smallest uncrossed off number left.4. If k is less than n then repeat process from step (2).5. Any number which has not been crossed off is a prime number.The Sieve of EratosthenesThe process is illustrated below:The word algorithm inEnglish comes from the Arab mathematician al-Khowarizmi who flourished at the end of the8th Century A.D. Al-Khowarizmi had reported the Indian discovery of the decimal number system withzero and introduced a number of important concepts in his book Al-jabr wa'l muqabala whose titlegives us the word algebra in English. Another Arab mathematician, al-Khashi (1393-1449), devised a method of computing the decimal expansion of pi (the ratio between a circle's circumference and its diameter)^4, obtaining a result accurate to 16 places.4) A method of memorising this is the phrase: `How I needa drink, alcoholic of course, after all those lectures involving tedious mnemonics for pi'.The number of letters in each word gives the sequence of digits in the expansion of pi.Al-Khashi also devised the first mechanical computing devices which could be used to predict the occurrence of important celestial phenomena such as solar and lunar eclipses.In summary, by the middle of the 15th Century there had evolved a flexible and powerful systemfor representing numerical quantities (both whole numbers and decimal fractions); some basic algorithmic techniques had been developed for manipulating these representations to deal with various calculations; andthe first moderately sophisticated aids to such calculations had been developed.One of the important trends in the history of this topic becomes apparent around this period: that asthe level of scientific and engineering knowledge increases, so in parallel does the complexity ofthe mathematical calculations associated with their development. Thus, to date, there has never beena point reached where the contemporary state of computational aids has exceeded the most demanding requirementsof contemporary sciences. We can cite two examples of this at the end of the 15th century. First,increasing knowledge about matters such asplanetary motion had resulted in data from observations that was increasingly difficult to process andinterpret with the methods available. Second, there was little in the way of devices to help withnavigation at a time when there was a considerable increase in explorationand commercial trade. In the next section of these notes we examine the continuing development ofcomputational mechanisms following this period.Previous SectionNext SectionPED Home Page Article Open access 03 February 2020 In the post-World War II years, computers became increasingly important for solving very difficult problems in mathematics, physics, chemistry, and other fields of science. By making it possible to find numerical solutions to equations that could not be solved analytically, computers helped to revolutionize many areas of scientific inquiry and engineering design. This trend continues as supercomputers are used to model weather systems, nuclear explosions, and airflow around new airplane designs. Since this trend has not yet leveled off, it is still too soon to say what the final result of these computational methods will be, but they have been revolutionary thus far and seem likely to become even more important in the future.BackgroundThe first attempts to invent devices to help with mathematical calculations date back at least 2,000 years, to the ancestor of the abacus. "Rod numerals," used in India and China, were not only used for counting, but also for simple calculations, too. These gave way to the abacus, which was used throughout Asia and beyond for several hundred years. However, in spite of the speed and accuracy with which addition, subtraction, multiplication, and division could be done on these devices, they were much less useful for the more complex mathematics that were being invented.The next step towards a mechanical calculator was taken in 1642 by Blaise Pascal (1623-1662), who developed a machine that could add numbers. Thirty years later, German mathematician Gottfried Leibniz (1646-1716) invented a machine that could perform all four basic arithmetic functions to help him avoid the tedium of manually calculating astronomical tables. In developing this machine, Leibniz stated, "it is unworthy of excellent men to lose hours like slaves in the labor of calculation which could safely be relegated to anyone else if machines were used." Unfortunately, mechanical inaccuracies in Leibniz's machineunavoidable, given the technology of the timesrendered it undependable for any but the simplest calculations.In 1822 English inventor Charles Babbage (1792-1871) developed a mechanical calculator called the "Difference Engine." Unlike previous machines, Babbage's invention could also solve some equations in addition to performing arithmetic operations. In later years, Babbage attempted to construct a more generalized machine, called an Analytical Engine, that could be programmed to do any mathematical operations. However, he failed to build it because of the technological limitations under which he worked. Others tried to carry Babbage's vision through to fruition, also without success because they, like he, were limited to constructing mechanical devices that had unavoidable inaccuracies built into them due to the technology of the day.With the development of electronics in the 1900s, the potential finally existed to construct an electronic machine to perform calculations. In the 1930s, electrical engineers were able to show that electromechanical circuits could be built that would add, subtract, multiply, and divide, finally bringing machines up to the level of the abacus. Pushed by the necessities of World War II, the Americans developed massive computers, the Mark I and ENIAC, to help solve ballistics problems for artillery shells, while the British, with their computer, Colossus, worked to break German codes. Meanwhile, English mathematician Alan Turing (1912-1954) was busy thinking about the next phase of computing, in which computers could be made to treat symbols the same as numbers and could be made to do virtually anything.Turing and his colleagues used their computers to help break German codes, helping to turn the tide of the Second World War in favor of the Allies. In the United States, simpler machines were used to help with the calculations under way in Los Alamos, where the first atomic bomb was under development. Meanwhile, in Boston and Aberdeen, Maryland, larger computers were working out ballistic problems. All of these efforts were of enormous importance toward the Allied victories over Germany and Japan, and proved the utility of the electronic computer to any doubters.The first purely scientific problem taken on by the electronic computers was that of solving Schroedinger's "wave equation," one of the central equations used in the then-new field of quantum mechanics. Although this equation, properly used, could provide exact solutions to many vexing problems in physics, it was so complex as to defy manual solution. Part of the reason for this involved the nature of the equation itself. For a simple atom, the number of calculations necessary to precisely show the locations and interactions of a single electron with its neighbors could be up to one million. For several electrons in the same atom, the number of calculations increased dramatically, pushing such problems beyond the range of even most of today's computers. Attacking the wave equation was one of the first tasks of the "newer" computers of the 1950s and 1960s, although it was not until the 1990s that supercomputers were available that could actually do a credible job of examining complex atoms or molecules.Through the 1960s and 1970s scientific computers became steadily more powerful, giving mathematicians, scientists, and engineers everbetter computational tools with which to ply their trades. However, these were invariably mainframe and "mini" computers because the personal computer and workstation had not yet been invented. This began to change in the 1980s with the introduction of the first affordable and (for that time) powerful small computers. With the advent of powerful personal computers, Sun and Silicon Graphics workstations, and other machines that could fit on a person's desk, scientific computing reached yet another level, especially in the late 1990s when desktop computers became nearly as powerful as the supercomputers of the 1970s and 1980s. At the same time, supercomputers continued to evolve, putting incredible amounts of computational power at the fingertips of researchers. Both of these trends continue to this day with no signs of abating.ImpactThe impact of computational methods of mathematics, science, and engineering has been nothing short of staggering. In particular, computers have made it possible to numerically solve important problems in mathematics, physics, and engineering that were hitherto unsolvable.One of the ways to solve a mathematical problem is to do so analytically. To solve a problem analytically, the mathematician will attempt, using only mathematical symbols and accepted mathematical operations, to come up with some answer that is a solution to the problem. For example, the problem 0 = x2 - 4 has an exact solution that can be arrived at analytically. By rewriting this problem as x2 = 4 (which is the result of adding 4 to each side of the equality), we can then determine that x is the square root of 4, which means that x = 2. This is an analytical solutionbecause it was arrived at by simple manipulations of the original equation using standard algebraic rules.On the other hand, more complex equations are not nearly so amenable to analytical solution. Schroedinger's wave equation, for example, cannot be solved in this manner because of its extreme complexity. Equations describing the flow of turbulent air past an airplane wing are similarly intractable, as are other problems in mathematics. However, these problems can be solved numerically, using computers.The simplest and least elegant way to solve a problem numerically is simply to program the computer to take a guess at a solution and, depending on whether the answer is too high or too low, to guess again with a larger or smaller number. This process repeats until the answer is found. In the above example, for instance, a computer would start at zero and would notice that (0)2 - 4 = -4. This answer is too small, so the computer would guess again. A second guess of 1 would give an answer of -3, still too small. Guessing 2 would make the equation work, ending the problem. Similarly, computers can be programmed to take this brute force approach with virtually any problem, returning numerical answers for nearly any equation that can be written.In other cases, for example, in calculating the flow of fluids, a computer will be programmed with the equations showing how a fluid behaves under certain conditions or at certain locations. It then systematically calculates the different parameters (for example, pressure, speed, and temperature) at hundreds or thousands of locations. Since each of these values will affect those around it (for example, a single hot point will tend to cool off as it warms neighboring points), the computer is also programmed to go back and recalculate all of these values, based on its first calculations. It repeats this process over and over until satisfied that the calculations are as accurate as they can be.Consider, for example, the problem of trying to calculate the temperatures all across a circuit board. If the temperature of any single point is the average of the four points adjacent to it, the computer will simply take those four points, average their temperatures, and give that value to the point in the middle. However, when this happens, the calculated temperature of all the surrounding points will change because now the central point has a different temperature. When they change, they in turn affect the central point again, and this cycle of calculations continues until the change in successive iterations is toosmall to matter much. This is called "finite difference" computation, and it is a powerful tool in the hands of engineers and scientists.The bottom line is that computer methods in the sciences have had an enormous impact on mathematics, the sciences, engineering, and our world. By freeing skilled scientists from the drudgery of endless calculations, they have freed these people to make more and more important discoveries in their fields. And by making some complex problems solvable for the first time, they have helped us to design better machines, to better understand our world and universe, and to make advances that would have otherwise been impossible.P. ANDREW KARAMFurther ReadingBanchoff, Thomas. Beyond the Third Dimension. New York: Scientific American Library, 1990.Kaufmann, William, and Larry Smarr. Supercomputing and the Transformation of Science. New York: Scientific American Library, 1993. Share copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. First published Mon Dec 18, 2000; substantive revision Fri Jun 9, 2006 Historically, computers were human clerks who calculated in accordancewith effective methods. These human computers did the sorts ofcalculation nowadays carried out by electronic computers, and manythousands of them were employed in commerce, government, and researchestablishments. The term computing machine, used increasinglyfrom the 1920s, refers to any machine that does the work of a humancomputer, i.e., any machine that calculates in accordance witheffective methods. During the late 1940s and early 1950s, with theadvent of electronic computing machines, the phrase computingmachine gradually gave way simply to computer,initially usually with the prefix electronic ordigital. This entry surveys the history of thesemachines. Charles Babbage was Lucasian Professor of Mathematics at CambridgeUniversity from 1828 to 1839 (a post formerly held by Isaac Newton).Babbage's proposed Difference Engine was a special-purpose digitalcomputing machine for the automatic production of mathematical tables(such as logarithm tables, tide tables, and astronomical tables). TheDifference Engine consisted entirely of mechanical components brass gear wheels, rods, ratchets, pinions, etc. Numbers wererepresented in the decimal system by the positions of 10-toothed metalwheels mounted in columns. Babbage exhibited a small working model in1822. He never completed the full-scale machine that he had designedbut did complete several fragments. The largest one ninth ofthe complete calculator is on display in the London ScienceMuseum. Babbage used it to perform serious computational work,calculating various mathematical tables. In 1990, Babbage's DifferenceEngine No. 2 was finally built from Babbage's designs and is also ondisplay at the London Science Museum. The Swedes Georg and Edvard Scheutz (father and son) constructed amodified version of Babbage's Difference Engine. Three were made, aprototype and two commercial models, one of these being sold to anobservatory in Albany, New York, and the other to theRegistrar-General's office in London, where it calculated and printedactuarial tables. Babbage's proposed Analytical Engine, considerably more ambitiousthan the Difference Engine, was to have been a general-purposemechanical digital computer. The Analytical Engine was to have had amemory store and a central processing unit (or mill) andwould have been able to select from among alternative actionsconsequent upon the outcome of its previous actions (a facilitynowadays known as conditional branching). The behaviour of theAnalytical Engine would have been controlled by a program ofinstructions contained on punched cards connected together with ribbons(an idea that Babbage had adopted from the Jacquard weaving loom).Babbage emphasised the generality of the Analytical Engine, sayingthe conditions which enable a finite machine to makecalculations of unlimited extent are fulfilled in the AnalyticalEngine (Babbage [1994], p. 97). Babbage worked closely with Ada Lovelace, daughter of the poetByron, after whom the modern programming language ADA is named.Lovelace foresaw the possibility of using the Analytical Engine fornon-numeric computation, suggesting that the Engine might even becapable of composing elaborate pieces of music. A large model of the Analytical Engine was under construction at thetime of Babbage's death in 1871 but a full-scale version was neverbuilt. Babbage's idea of a general-purpose calculating engine was neverforgotten, especially at Cambridge, and was on occasion a lively topicof mealtime discussion at the war-time headquarters of the GovernmentCode and Cypher School, Bletchley Park, Buckinghamshire, birthplace ofthe electronic digital computer. Analog computers The earliest computing machines in wide use were not digital butanalog. In analog representation, properties of the representationalmedium ape (or reflect or model) properties of the representedstate-of-affairs. (In obvious contrast, the strings of binary digitsemployed in digital representation do not represent by meansof possessing some physical property such as length whose magnitude varies in proportion to the magnitude of the propertythat is being represented.) Analog representations form a diverseclass. Some examples: the longer a line on a road map, the longer theroad that the line represents; the greater the number of clear plasticsquares in an architect's model, the greater the number of windows inthe building represented; the higher the pitch of an acoustic depthmeter, the shallower the water. In analog computers, numericalquantities are represented by, for example, the angle of rotation of ashaft or a difference in electrical potential. Thus the output voltageof the machine at a time might represent the momentary speed of theobject being modelled. As the case of the architect's model makes plain, analogrepresentation may be discrete in nature (there is no suchthing as a fractional number of windows). Among computer scientists,the term analog is sometimes used narrowly, to indicaterepresentation of one continuously-valued quantity by another(e.g., speed by voltage). As Brian Cantwell Smith has remarked: Analog should be a predicate on arepresentation whose structure corresponds to that of which itrepresents That continuous representations should historicallyhave come to be called analog presumably betrays the recognition that,at the levels at which it matters to us, the world is morefoundationally continuous than it is discrete. (Smith [1991], p.271) James Thomson, brother of Lord Kelvin, invented the mechanicalwheel-and-disc integrator that became the foundation of analogcomputation (Thomson [1876]). The two brothers constructed a device forcomputing the integral of the product of two given functions, andKelvin described (although did not construct) general-purpose analogmachines for integrating linear differential equations of any order andfor solving simultaneous linear equations. Kelvin's most successfulanalog computer was his tide predicting machine, which remained in useat the port of Liverpool until the 1960s. Mechanical analog devicesbased on the wheel-and-disc integrator were in use during World War Ifor gunnery calculations. Following the war, the design of theintegrator was considerably improved by Hannibal Ford (Ford [1919]). Stanley Fifer reports that the first semi-automatic mechanicalanalog computer was built in England by the Manchester firm ofMetropolitan Vickers prior to 1930 (Fifer [1961], p. 29); however, Ihave so far been unable to verify this claim. In 1931, Vannevar Bush,working at MIT, built the differential analyser, the first large-scaleautomatic general-purpose mechanical analog computer. Bush's design wasbased on the wheel and disc integrator. Soon copies of his machine werein use around the world (including, at Cambridge and ManchesterUniversities in England, differential analysers built out of kit-setMeccano, the once popular engineering toy). It required a skilled mechanic equipped with a lead hammer to set upBush's mechanical differential analyser for each new job. Subsequently,Bush and his colleagues replaced the wheel-and-disc integrators andother mechanical components by electromechanical, and finally byelectronic, devices. A differential analyser may be conceptualised as a collection ofblack boxes connected together in such a way as to allowconsiderable feedback. Each box performs a fundamental process, forexample addition, multiplication of a variable by a constant, andintegration. In setting up the machine for a given task, boxes areconnected together so that the desired set of fundamental processes isexecuted. In the case of electrical machines, this was done typicallyby plugging wires into sockets on a patch panel (computing machineswhose function is determined in this way are referred to asprogram-controlled). Since all the boxes work in parallel, an electronic differentialanalyser solves sets of equations very quickly. Against this has to beset the cost of massaging the problem to be solved into the formdemanded by the analog machine, and of setting up the hardware toperform the desired computation. A major drawback of analog computationis the higher cost, relative to digital machines, of an increase inprecision. During the 1960s and 1970s, there was considerable interestin hybrid machines, where an analog section is controlledby and programmed via a digital section. However, such machines are nowa rarity. The Universal Turing Machine In 1936, at Cambridge University, Turing invented the principle of themodern computer. He described an abstract digital computing machineconsisting of a limitless memory and a scanner that moves back andforth through the memory, symbol by symbol, reading what it finds andwriting further symbols (Turing [1936]). The actions of the scanner aredictated by a program of instructions that is stored in the memory inthe form of symbols. This is Turing's stored-program concept, andimplicit in it is the possibility of the machine operating on andmodifying its own program. (In London in 1947, in the course of whatwas, so far as is known, the earliest public lecture to mentioncomputer intelligence, Turing said, What we want is a machinethat can learn from experience, adding that thepossibility of letting the machine alter its own instructionsprovides the mechanism for this (Turing [1947] p. 393). Turing'scomputing machine of 1936 is now known simply as the universal Turingmachine. Cambridge mathematician Max Newman remarked that right fromthe start Turing was interested in the possibility of actually buildinga computing machine of the sort that he had described (Newman ininterview with Christopher Evans in Evans [197?]. From the start of the Second World War Turing was a leadingcryptanalyst at the Government Code and Cypher School, Bletchley Park.Here he became familiar with Thomas Flowers' work involving large-scalehigh-speed electronic switching (described below). However, Turingcould not turn to the project of building an electronic stored-programcomputing machine until the cessation of hostilities in Europe in1945. During the wartime years Turing did give considerable thought to thequestion of machine intelligence. Colleagues at Bletchley Park recallnumerous off-duty discussions with him on the topic, and at one pointTuring circulated a typewritten report (now lost) setting out some ofhis ideas. One of these colleagues, Donald Michie (who later foundedthe Department of Machine Intelligence and Perception at the Universityof Edinburgh), remembers Turing talking often about the possibility ofcomputing machines (1) learning from experience and (2) solvingproblems by means of searching through the space of possible solutions,guided by rule-of-thumb principles (Michie in interview with Copeland,1995). The modern term for the latter idea is heuristicsearch, a heuristic being any rule-of-thumb principle that cutsdown the amount of searching required in order to find a solution to aproblem. At Bletchley Park Turing illustrated his ideas on machineintelligence by reference to chess. Michie recalls Turing experimentingwith heuristics that later became common in chess programming (inparticular minimax and best-first). Further information about Turing and the computer, including hiswartime work on codebreaking and his thinking about artificialintelligence and artificial life, can be found in Copeland 2004. Electromechanical versus Electronic Computation With some exceptions including Babbage's purely mechanicalengines, and the finger-powered National Accounting Machine - earlydigital computing machines were electromechanical. That is to say,their basic components were small, electrically-driven, mechanicalswitches called relays. These operate relatively slowly,whereas the basic components of an electronic computer originally vacuum tubes (valves) have no moving parts saveelectrons and so operate extremely fast. Electromechanical digitalcomputing machines were built before and during the second world war by(among others) Howard Aiken at Harvard University, George Stibitz atBell Telephone Laboratories, Turing at Princeton University andBletchley Park, and Konrad Zuse in Berlin. To Zuse belongs the honourof having built the first working general-purpose program-controlleddigital computer. This machine, later called the Z3, was functioning in1941. (A program-controlled computer, as opposed to a stored-programcomputer, is set up for a new task by re-routing wires, by means ofplugs etc.) Relays were too slow and unreliable a medium for large-scalegeneral-purpose digital computation (although Aiken made a valianteffort). It was the development of high-speed digital techniques usingvacuum tubes that made the modern computer possible. The earliest extensive use of vacuum tubes for digitaldata-processing appears to have been by the engineer Thomas Flowers,working in London at the British Post Office Research Station at DollisHill. Electronic equipment designed by Flowers in 1934, for controllingthe connections between telephone exchanges, went into operation in1939, and involved between three and four thousand vacuum tubes runningcontinuously. In 19381939 Flowers worked on an experimentalelectronic digital data-processing system, involving a high-speed datastore. Flowers' aim, achieved after the war, was that electronicequipment should replace existing, less reliable, systems built fromrelays and used in telephone exchanges. Flowers did not investigate theidea of using electronic equipment for numerical calculation, but hasremarked that at the outbreak of war with Germany in 1939 he waspossibly the only person in Britain who realized that vacuum tubescould be used on a large scale for high-speed digital computation. (SeeCopeland 2006 for m more information on Flowers' work.) Atanasoff The earliest comparable use of vacuum tubes in the U.S. seems to havebeen by John Atanasoff at what was then Iowa State College (nowUniversity). During the period 19371942 Atanasoff developedtechniques for using vacuum tubes to perform numerical calculationsdigitally. In 1939, with the assistance of his student Clifford Berry,Atanasoff began building what is sometimes called the Atanasoff-BerryComputer, or ABC, a small-scale special-purpose electronic digitalmachine for the solution of systems of linear algebraic equations. Themachine contained approximately 300 vacuum tubes. Although theelectronic part of the machine functioned successfully, the computer asa whole never worked reliably, errors being introduced by theunsatisfactory binary card-reader. Work was discontinued in 1942 whenAtanasoff left Iowa State. Colossus The first fully functioning electronic digital computer was Colossus,used by the Bletchley Park cryptanalysts from February 1944. From very early in the war the Government Code and Cypher School(GC&CS) was successfully deciphering German radio communicationsencoded by means of the Enigma system, and by early 1942 about 39,000intercepted messages were being decoded each month, thanks toelectromechanical machines known as bombes. These weredesigned by Turing and Gordon Welchman (building on earlier work byPolish cryptanalysts). During the second half of 1941, messages encoded by means of atotally different method began to be intercepted. This new ciphermachine, code-named Tunny by Bletchley Park, was brokenin April 1942 and current traffic was read for the first time in Julyof that year. Based on binary teleprinter code, Tunny was used inpreference to Morse-based Enigma for the encryption of high-levelsignals, for example messages from Hitler and members of the GermanHigh Command. The need to decipher this vital intelligence as rapidly as possibleled Max Newman to propose in November 1942 (shortly after hisrecruitment to GC&CS from Cambridge University) that key parts ofthe decryption process be automated, by means of high-speed electroniccounting devices. The first machine designed and built to Newman'sspecification, known as the Heath Robinson, was relay-based withelectronic circuits for counting. (The electronic counters weredesigned by C.E. Wynn-Williams, who had been using thyratron tubes incounting circuits at the Cavendish Laboratory, Cambridge, since 1932[Wynn-Williams 1932].) Installed in June 1943, Heath Robinson wasunreliable and slow, and its high-speed paper tapes were continuallybreaking, but it proved the worth of Newman's idea. Flowers recommendedthat an all-electronic machine be built instead, but he received noofficial encouragement from GC&CS. Working independently at thePost Office Research Station at Dollis Hill, Flowers quietly got onwith constructing the world's first large-scale programmable electronicdigital computer. Colossus I was delivered to Bletchley Park in January1943. By the end of the war there were ten Colossi working round the clockat Bletchley Park. From a cryptanalytic viewpoint, a major differencebetween the prototype Colossus I and the later machines was theaddition of the so-called Special Attachment, following a key discoveryby cryptanalysts Donald Michie and Jack Good. This broadened thefunction of Colossus from wheel setting i.e.,determining the settings of the encoding wheels of the Tunny machinefor a particular message, given the patterns of thewheels to wheel breaking, i.e., determining thewheel patterns themselves. The wheel patterns were eventually changeddaily by the Germans on each of the numerous links between the GermanArmy High Command and Army Group commanders in the field. By 1945 therewere as many 30 links in total. About ten of these were broken and readregularly. Colossus I contained approximately 1600 vacuum tubes and each of thesubsequent machines approximately 2400 vacuum tubes. Like the smallerABC, Colossus lacked two important features of modern computers. First,it had no internally stored programs. To set it up for a new task, theoperator had to alter the machine's physical wiring, using plugs andswitches. Second, Colossus was not a general-purpose machine, beingdesigned for a specific cryptanalytic task involving counting andBoolean operations. F.H. Hinsley, official historian of GC&CS, has estimated thatthe war in Europe was shortened by at least two years as a result ofthe signals intelligence operation carried out at Bletchley Park, inwhich Colossus played a major role. Most of the Colossi were destroyedonce hostilities ceased. Some of the electronic panels ended up atNewman's Computing Machine Laboratory in Manchester (see below), alltrace of their original use having been removed. Two Colossi wereretained by GC&CS (renamed GCHQ following the end of the war). Thelast Colossus is believed to have stopped running in 1960. Those who knew of Colossus were prohibited by the Official SecretsAct from sharing their knowledge. Until the 1970s, few had any ideathat electronic computation had been used successfully during thesecond world war. In 1970 and 1975, respectively, Good and Michiepublished not

numbers,say in the scale of 2, so that each number appears as a row of, say,forty 0's and 1's in certain places or "houses" in the machine. Certain of these numbers, or "words" are read, one afteranother, as orders. In one possible type of machine an order consistsof four numbers, for example 11, 13, 27, 4. The number 4 signifies"add", and when control shifts to this word the "houses" H11 and H13will be connected to the adder as inputs, and H27 as output. Thenumbers stored in H11 and H13 pass through the adder, are added, andthe sum is passed on to H27. The control then shifts to the next order.In most real machines the process just described would be done by threeseparate orders, the first bringing [H11] (=content of H11) to acentral accumulator, the second adding [H13] into the accumulator, andthe third sending the result to H27; thus only one address would berequired in each order. A machine with storage, with thisautomatic-telephone-exchange arrangement and with the necessary adders,subtractors and so on, is, in a sense, already a universal machine.(Newman [1948], pp. 271272) Following this explanation of Turing's three-address concept (source1, source 2, destination, function) Newman went on to describe programstorage (the orders shall be in a series of houses X1, X2,) and conditional branching. He then summed up: From this highly simplified account it emerges that theessential internal parts of the machine are, first, a storage fornumbers (which may also be orders). Secondly, adders,multipliers, etc. Thirdly, an "automatic telephone exchange" forselecting "houses", connecting them to the arithmetic organ, andwriting the answers in other prescribed houses. Finally, means ofmoving control at any stage to any chosen order, if a certain conditionis satisfied, otherwise passing to the next order in the normalsequence. Besides these there must be ways of setting up the machine atthe outset, and extracting the final answer in useable form. (Newman[1948], pp. 2734) In a letter written in 1972 Williams described in some detail whathe and Kilburn were told by Newman: About the middle of the year [1946] the possibility of anappointment at Manchester University arose and I had a talk withProfessor Newman who was already interested in the possibility ofdeveloping computers and had acquired a grant from the Royal Society of30,000 for this purpose. Since he understood computers and Iunderstood electronics the possibilities of fruitful collaboration wereobvious. I remember Newman giving us a few lectures in which heoutlined the organisation of a computer in terms of numbers beingidentified by the address of the house in which they were placed and interms of numbers being transferred from this address, one at a time, toan accumulator where each entering number was added to what was alreadythere. At any time the number in the accumulator could be transferredback to an assigned address in the store and the accumulator clearedfor further use. The transfers were to be effected by a stored programin which a list of instructions was obeyed sequentially. Orderedprogress through the list could be interrupted by a test instructionwhich examined the sign of the number in the accumulator. Thereafteroperation started from a new point in the list of instructions. Thiswas the first information I received about the organisation ofcomputers. Our first computer was the simplest embodiment ofthese principles, with the sole difference that it used a subtractingrather than an adding accumulator. (Letter from Williams to Randell,1972; in Randell [1972], p. 9) Turing's early input to the developments at Manchester, hinted at byWilliams in his above-quoted reference to Turing, may have been via thelectures on computer design that Turing and Wilkinson gave in Londonduring the period December 1946 to February 1947 (Turing and Wilkinson[19467]). The lectures were attended by representatives ofvarious organisations planning to use or build an electronic computer.Kilburn was in the audience (Bowker and Giordano [1993]). (Kilburnusually said, when asked from where he obtained his basic knowledge ofthe computer, that he could not remember (letter from Brian Napper toCopeland, 2002); for example, in a 1992 interview he said:Between early 1945 and early 1947, in that period, somehow orother I knew what a digital computer was Where I got thisknowledge from I've no idea (Bowker and Giordano [1993], p.19).) Whatever role Turing's lectures may have played in informingKilburn, there is little doubt that credit for the Manchester computer called the Newman-Williams machine in acontemporary document (Huskey 1947) belongs not only toWilliams and Kilburn but also to Newman, and that the influence onNewman of Turing's 1936 paper was crucial, as was the influence ofFlowers' Colossus. The first working AI program, a draughts (checkers) player writtenby Christopher Strachey, ran on the Ferranti Mark I in the ManchesterComputing Machine Laboratory. Strachey (at the time a teacher at HarrowSchool and an amateur programmer) wrote the program with Turing'sencouragement and utilising the latter's recently completedProgrammers' Handbook for the Ferranti. (Strachey later became Directorof the Programming Research Group at Oxford University.) By the summerof 1952, the program could, Strachey reported, play a completegame of draughts at a reasonable speed. (Strachey's program formed thebasis for Arthur Samuel's well-known checkers program.) The firstchess-playing program, also, was written for the Manchester Ferranti,by Dietrich Prinz; the program first ran in November 1951. Designed forsolving simple problems of the mate-in-two variety, the program wouldexamine every possible move until a solution was found. Turing startedto program his Turochamp chess-player on the FerrantiMark I, but never completed the task. Unlike Prinz's program, theTurochamp could play a complete game (when hand-simulated) and operatednot by exhaustive search but under the guidance of heuristics. ENIAC and EDVAC The first fully functioning electronic digital computer to be built inthe U.S. was ENIAC, constructed at the Moore School of ElectricalEngineering, University of Pennsylvania, for the Army OrdnanceDepartment, by J. Presper Eckert and John Mauchly. Completed in 1945,ENIAC was somewhat similar to the earlier Colossus, but considerablylarger and more flexible (although far from general-purpose). Theprimary function for which ENIAC was designed was the calculation oftables used in aiming artillery. ENIAC was not a stored-programcomputer, and setting it up for a new job involved reconfiguring themachine by means of plugs and switches. For many years, ENIAC wasbelieved to have been the first functioning electronic digitalcomputer, Colossus being unknown to all but a few. In 1944, John von Neumann joined the ENIAC group. He had becomeintrigued (Goldstine's word, [1972], p. 275) withTuring's universal machine while Turing was at Princeton Universityduring 19361938. At the Moore School, von Neumann emphasised theimportance of the stored-program concept for electronic computing,including the possibility of allowing the machine to modify its ownprogram in useful ways while running (for example, in order to controlloops and branching). Turing's paper of 1936 (On ComputableNumbers, with an Application to the Entscheidungsproblem) wasrequired reading for members of von Neumann's post-war computer projectat the Institute for Advanced Study, Princeton University (letter fromJulian Bigelow to Copeland, 2002; see also Copeland [2004], p. 23).Eckert appears to have realised independently, and prior to vonNeumann's joining the ENIAC group, that the way to take full advantageof the speed at which data is processed by electronic circuits is toplace suitably encoded instructions for controlling the processing inthe same high-speed storage devices that hold the data itself(documented in Copeland [2004], pp. 267). In 1945, while ENIACwas still under construction, von Neumann produced a draft report,mentioned previously, setting out the ENIAC group's ideas for anelectronic stored-program general-purpose digital computer, the EDVAC(von Neuman [1945]). The EDVAC was completed six years later, but notby its originators, who left the Moore School to build computerselsewhere. Lectures held at the Moore School in 1946 on the proposedEDVAC were widely attended and contributed greatly to the disseminationof the new ideas. Von Neumann was a prestigious figure and he made the concept of ahigh-speed stored-program digital computer widely known through hiswritings and public addresses. As a result of his high profile in thefield, it became customary, although historically inappropriate, torefer to electronic stored-program digital computers as vonNeumann machines. The Los Alamos physicist Stanley Frankel, responsible with vonNeumann and others for mechanising the large-scale calculationsinvolved in the design of the atomic bomb, has described von Neumann'sview of the importance of Turing's 1936 paper, in a letter: I know that in or about 1943 or 44 von Neumann waswell aware of the fundamental importance of Turing's paper of 1936 Von Neumann introduced me to that paper and at his urging Istudied it with care. Many people have acclaimed von Neumann as the"father of the computer" (in a modern sense of the term) but I am surethat he would never have made that mistake himself. He might well becalled the midwife, perhaps, but he firmly emphasized to me, and toothers I am sure, that the fundamental conception is owing to Turing,in so far as not anticipated by Babbage Both Turing and vonNeumann, of course, also made substantial contributions to the"reduction to practice" of these concepts but I would not regard theseas comparable in importance with the introduction and explication ofthe concept of a computer able to store in its memory its program ofactivities and of modifying that program in the course of theseactivities. (Quoted in Randell [1972], p. 10) Other Notable Early Computers Other notable early stored-program electronic digital computers were: EDSAC, 1949, built at Cambridge University by Maurice Wilkes BINAC, 1949, built by Eckert's and Mauchly's Electronic ControlCo., Philadelphia (opinions differ over whether BINAC ever actuallyworked) Whirlwind I, 1949, Digital Computer Laboratory, MassachusettsInstitute of Technology, Jay Forrester SEAC, 1950, US Bureau of Standards Eastern Division, WashingtonD.C., Samuel Alexander, Ralph Slutz SWAC, 1950, US Bureau of Standards Western Division, Institute forNumerical Analysis, University of California at Los Angeles, HarryHuskey UNIVAC, 1951, Eckert-Mauchly Computer Corporation, Philadelphia(the first computer to be available commercially in the U.S.) the IAS computer, 1952, Institute for Advanced Study, PrincetonUniversity, Julian Bigelow, Arthur Burks, Herman Goldstine, vonNeumann, and others (thanks to von Neumann's publishing thespecifications of the IAS machine, it became the model for a group ofcomputers known as the Princeton Class machines; the IAS computer wasalso a strong influence on the IBM 701) IBM 701, 1952, International Business Machine's first mass-producedelectronic stored-program computer. High-Speed Memory The EDVAC and ACE proposals both advocated the use of mercury-filledtubes, called delay lines, for high-speed internalmemory. This form of memory is known as acoustic memory. Delay lineshad initially been developed for echo cancellation in radar; the ideaof using them as memory devices originated with Eckert at the MooreSchool. Here is Turing's description: It is proposed to build "delay line" units consisting ofmercury tubes about 5 long and 1 in diameter incontact with a quartz crystal at each end. The velocity of sound in mercury is such that the delay will be 1.024 ms. Theinformation to be stored may be considered to be a sequence of 1024digits (0 or 1) These digits will be representedby a corresponding sequence of pulses. The digit 0 will berepresented by the absence of a pulse at the appropriate time, thedigit 1 by its presence. This series of pulses is impressed onthe end of the line by one piezo-crystal, it is transmitted down theline in the form of supersonic waves, and is reconverted into a varyingvoltage by the crystal at the far end. This voltage is amplifiedsufficiently to give an output of the order of 10 volts peak to peakand is used to gate a standard pulse generated by the clock. This pulsemay be again fed into the line by means of the transmitting crystal, orwe may feed in some altogether different signal. We also have thepossibility of leading the gated pulse to some other part of thecalculator, if we have need of that information at the time. Making useof the information does not of course preclude keeping it also. (Turing[1945], p. 375) Mercury delay line memory was used in EDSAC, BINAC, SEAC, PilotModel ACE, EDVAC, DEUCE, and full-scale ACE (1958). The chief advantageof the delay line as a memory medium was, as Turing put it, that delaylines were "already a going concern" (Turing [1947], p. 380). Thefundamental disadvantages of the delay line were that random access isimpossible and, moreover, the time taken for an instruction, or number,to emerge from a delay line depends on where in the line it happens tobe. In order to minimize waiting-time, Turing arranged for instructionsto be stored not in consecutive positions in the delay line, but inrelative positions selected by the programmer in such a way that eachinstruction would emerge at exactly the time it was required, in so faras this was possible. Each instruction contained a specification of thelocation of the next. This system subsequently became known asoptimum coding. It was an integral feature of everyversion of the ACE design. Optimum coding made for difficult and untidyprogramming, but the advantage in terms of speed was considerable.Thanks to optimum coding, the Pilot Model ACE was able to do a floatingpoint multiplication in 3 milliseconds (Wilkes's EDSAC required 4.5milliseconds to perform a single fixed point multiplication). In the Williams tube or electrostatic memory, previously mentioned,a two-dimensional rectangular array of binary digits was stored on theface of a commercially-available cathode ray tube. Access to data wasimmediate. Williams tube memories were employed in the Manchesterseries of machines, SWAC, the IAS computer, and the IBM 701, and amodified form of Williams tube in Whirlwind I (until replacement bymagnetic core in 1953). Drum memories, in which data was stored magnetically on the surfaceof a metal cylinder, were developed on both sides of the Atlantic. Theinitial idea appears to have been Eckert's. The drum providedreasonably large quantities of medium-speed memory and was used tosupplement a high-speed acoustic or electrostatic memory. In 1949, theManchester computer was successfully equipped with a drum memory; thiswas constructed by the Manchester engineers on the model of a drumdeveloped by Andrew Booth at Birkbeck College, London. The final major event in the early history of electronic computationwas the development of magnetic core memory. Jay Forrester realisedthat the hysteresis properties of magnetic core (normally used intransformers) lent themselves to the implementation of athree-dimensional solid array of randomly accessible storage points. In1949, at Massachusetts Institute of Technology, he began to investigatethis idea empirically. Forrester's early experiments with metallic coresoon led him to develop the superior ferrite core memory. DigitalEquipment Corporation undertook to build a computer similar to theWhirlwind I as a test vehicle for a ferrite core memory. The MemoryTest Computer was completed in 1953. (This computer was used in 1954for the first simulations of neural networks, by Belmont Farley andWesley Clark of MIT's Lincoln Laboratory (see Copeland and Proudfoot[1996]). Once the absolute reliability, relative cheapness, high capacity andpermanent life of ferrite core memory became apparent, core soonreplaced other forms of high-speed memory. The IBM 704 and 705computers (announced in May and October 1954, respectively) broughtcore memory into wide use. This is a timeline of key developments in computational mathematics.Monte Carlo simulation (voted one of the top 10 algorithms of the 20th century) invented at Los Alamos by von Neumann, Ulam and Metropolis.[1][2][3]Dantzig introduces the simplex algorithm (voted one of the top 10 algorithms of the 20th century).[4]First hydro simulations at Los Alamos occurred.[5][6]Ulam and von Neumann introduce the notion of cellular automata.[7]A routine for the Manchester Baby written to factor a large number (2^18), one of the first in computational number theory.[8] The Manchester group would make several other breakthroughs in this area.[9]LU decomposition technique first discovered.Hestenes, Stiefel, and Lanczos, all from the Institute for Numerical Analysis at the National Bureau of Standards, initiate the development of Krylov subspace iteration methods.[10][11][12][13] Voted one of the top 10 algorithms of the 20th century.Equations of State Calculations by Fast Computing Machines introduces the MetropolisHastings algorithm.[14] Also, important earlier independent work by Alder and S. Frankel.[15][16]Enrico Fermi, Stanislaw Ulam, John Pasta, and Mary Tsingou, discover the FermiPastaUlamTsingou problem.[17]In network theory, Ford & Fulkerson compute a solution to the maximum flow problem.[18]Householder invents his eponymous matrices and transformation method (voted one of the top 10 algorithms of the 20th century).[19]Molecular dynamics invented by Alder and Wainwright[20]John G.F. Francis[21] and Vera Kublanovskaya[22] invent QR factorization (voted one of the top 10 algorithms of the 20th century).First recorded use of the term "finite element method" by Ray Clough,[23] to describe the methods of Courant, Hrenikoff and Zienkiewicz, among others. See also here.Using computational investigations of the 3-body problem, Minovitch formulates the gravity assist method.[24][25]Molecular dynamics was invented independently by Aneesur Rahman.[26]Cooley and Tukey re-invent the Fast Fourier transform (voted one of the top 10 algorithms of the 20th century), an algorithm first discovered by Gauss.Edward Lorenz discovers the butterfly effect on a computer, attracting interest in chaos theory.[27]Kruskal and Zabusky follow up the FermiPastaUlamTsingou problem with further numerical experiments, and coin the term "soliton".[28][29]Birch and Swinnerton-Dyer conjecture formulated through investigations on a computer.[30]Grobner bases and Buchberger's algorithm invented for algebra[31]Frenchman Verlet (re)discovers a numerical integration algorithm,[32] (first used in 1791 by Delambre, by Cowell and Crommelin in 1909, and by Carl Fredrik Strmer in 1907,[33] hence the alternative names Strmer's method or the Verlet-Strmer method) for dynamics.[32]Risch invents algorithm for symbolic integration.[34]Mandelbrot, from studies of the Fatou, Julia and Mandelbrot sets, coined and popularized the term 'fractal' to describe these structures' self-similarity.[35][36]Kenneth Appel and Wolfgang Haken prove the four colour theorem, the first theorem to be proved by computer.[37][38][39]Fast multipole method invented by Rokhlin and Greengard (voted one of the top 10 algorithms of the 20th century).[40][41][42]The appearance of the first research grids using volunteer computing GIMPS (1996) and distributed.net (1997).Kepler conjecture is almost all but certainly proved algorithmically by Thomas Hales in 1998.In computational group theory, God's Number for the Rubik's cube is shown to be 20.[43][44]Mathematicians completely map the E8-group.[45][46][47]Hales completes the proof of Kepler's conjecture.[48][49][50]Timeline of scientific computingComputational mathematicsTimeline of algorithmsTimeline of mathematics from the 20th century onwardsTimeline of numerical analysis after 1945Portals: Science Mathematics^ Metropolis, N. (1987). "The Beginning of the Monte Carlo method" (PDF). Los Alamos Science. 15: 125130. Retrieved 5 May 2012.^ S. Ulam, R. D. Richtmyer, and J. von Neumann(1947). Statistical methods in neutron diffusion. Los Alamos Scientific Laboratory report LAMS551.^ N. Metropolis and S. Ulam (1949). The Monte Carlo method. Journal of the American Statistical Association 44:335341.^ "SIAM News, November 1994". Retrieved 6 June 2012. Systems Optimization Laboratory, Stanford University Huang Engineering Center (site host/mirror).^ Richtmyer, R. D. (1948). Proposed Numerical Method for Calculation of Shocks. Los Alamos, NM: Los Alamos Scientific Laboratory LA-671.^ A Method for the Numerical Calculation of Hydrodynamic Shocks.Von Neumann, J.; Richtmyer, R. D. Journal of Applied Physics, Vol. 21, pp. 232237^ Von Neumann, J., Theory of Self-Reproducing Automata, Univ. of Illinois Press, Urbana, 1966.^ The Manchester Mark 1.^ One tonne 'Baby' marks its birth: Dashing times. By Jonathan Fildes, Science and technology reporter, BBC News.^ Magnus R. Hestenes and Eduard Stiefel, Methods of Conjugate Gradients for Solving Linear Systems, J. Res. Natl. Bur. Stand. 49, 409436 (1952).^ Eduard Stiefel, U ber einige Methoden der Relaxationsrechnung (in German), Z. Angew. Math. Phys. 3, 133 (1952).^ Cornelius Lanczos, Solution of Systems of Linear Equations by Minimized Iterations, J. Res. Natl. Bur. Stand. 49, 3353 (1952).^ Cornelius Lanczos, An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, J. Res. Natl. Bur. Stand. 45, 255282 (1950).^ Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. (1953). "Equations of State Calculations by Fast Computing Machines". Journal of Chemical Physics. 21 (6): 10871092. Bibcode:1953JChPh..21.1087M. doi:10.1063/1.1699114. OSTI4390578. S2CID1046577.^ Unfortunately, Alder's thesis advisor was unimpressed, so Alder and Frankel delayed publication of their results until much later. Alder, B. J., Frankel, S. P., and Lewinson, B. A., J. Chem. Phys., 23, 3 (1955).^ Stanley P. Frankel, Unrecognized Genius, HP9825.COM (accessed 29 Aug 2015).^ Fermi, E. (posthumously); Pasta, J.; Ulam, S. (1955): Studies of Nonlinear Problems (accessed 25 Sep 2012). Los Alamos Laboratory Document LA-1940. Also appeared in 'Collected Works of Enrico Fermi', E. Segre ed., University of Chicago Press, Vol. II, 978988, 1965. Recovered 21 Dec 2012^ Ford, L. R.; Fulkerson, D. R. (1956). "Maximal flow through a network" . Canadian Journal of Mathematics. 8: 399404.^ Householder, A. S. (1958). "Unitary Triangularization of a Nonsymmetric Matrix" (PDF). Journal of the ACM. 5 (4): 339342. doi:10.1145/320941.320947. MR0111128. S2CID9858625.^ Alder, B. J.; T. E. Wainwright (1959). "Studies in Molecular Dynamics. I. General Method". J. Chem. Phys. 31 (2): 459. Bibcode 1959JChPh..31..459A. doi:10.1063/1.1730376^ J. G. F. Francis, "The QR Transformation, I", The Computer Journal, vol. 4, no. 3, pages 265271 (1961, received Oct 1959) online at oxfordjournals.org;J. G. F. Francis, "The QR Transformation, II" The Computer Journal, vol. 4, no. 4, pages 332345 (1962) online at oxfordjournals.org.^ Vera N. Kublanovskaya (1961), "On some algorithms for the solution of the complete eigenvalue problem," USSR Computational Mathematics and Mathematical Physics, 1(3), pages 637657 (1963, received Feb 1961). Also published in: Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki [Journal of Computational Mathematics and Mathematical Physics], 1(4), pages 555570 (1961).^ RW Clough, The Finite Element Method in PlaneStress Analysis, Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh, PA, Sept. 8, 9, 1960.^ Minovitch, Michael: "A method for determining interplanetary free-fall reconnaissance trajectories," Jet Propulsion Laboratory Technical Memo TM-312-130, pages 38-44 (23 August 1961).^ Christopher Riley and Dallas Campbell, Oct 22, 2012. "The maths that made Voyager possible" Archived 2013-07-30 at the Wayback Machine. BBC News Science and Environment. Recovered 16 Jun 2013.^ Rahman, A (1964). "Correlations in the Motion of Atoms in Liquid Argon". Phys Rev. 136 (2A): A405 A41. Bibcode:1964PhRv..136..405R. doi:10.1103/PhysRev.136.A405.^ Lorenz, Edward N. (1963). "Deterministic Nonperiodic Flow" (PDF). Journal of the Atmospheric Sciences. 20 (2): 130141. Bibcode:1963JAtS...20..130L. doi:10.1175/1520-0469(1963)0202.0.CO;2.^ Zabusky, N. J.; Kruskal, M. D. (1965). "Interaction of 'solitons' in a collisionless plasma and the recurrence of initial states". Phys. Rev. Lett. 15 (6): 240243. Bibcode 1965PhRvL..15..240Z. doi:10.1103/PhysRevLett.15.240.^ retrieved 3 nov 2012.^ Birch, Bryan; Swinnerton-Dyer, Peter (1965). "Notes on Elliptic Curves (II)". J. Reine Angew. Math. 165 (218): 79108. doi:10.1515/crll.1965.218.79.^ Bruno Buchberger: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (PDF; 1,8 MB). 1965^ a b Verlet, Loup (1967). "Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of LennardJones Molecules". Physical Review. 159 (1): 98103. Bibcode:1967PhRv..159...98V. doi:10.1103/PhysRev.159.98.^ Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 17.4. Second-Order Conservative Equations". Numerical Recipes: The Art of Scientific Computing (3rded.). New York: Cambridge University Press. ISBN978-0-521-88068-8.^ Risch, R. H. (1969). "The problem of integration in finite terms". Transactions of the American Mathematical Society. American Mathematical Society. 139: 167189. doi:10.2307/1995313. JSTOR 1995313.Risch, R. H. (1970). "The solution of the problem of integration in finite terms". Bulletin of the American Mathematical Society. 76 (3): 605608. doi:10.1090/S0002-9904-1970-12454-5.^ B. Mandelbrot; Les objets fractals, forme, hasard et dimension (in French). Publisher: Flammarion (1975), ISBN9782082106474; English translation Fractals: Form, Chance and Dimension. Publisher: Freeman, W. H & Company. (1977). ISBN9780716704737.^ Mandelbrot, Benot B.; (1983). The Fractal Geometry of Nature. San Francisco: W.H. Freeman. ISBN0-7167-1186-9.^ Kenneth Appel and Wolfgang Haken, "Every planar map is four colorable, Part I: Discharging," Illinois Journal of Mathematics 21: 429490, 1977.^ Appel, K. and Haken, W. "Every Planar Map is Four-Colorable, II: Reducibility." Illinois J. Math. 21, 491567, 1977.^ Appel, K. and Haken, W. "The Solution of the Four-Color Map Problem." Sci. Amer. 237, 108121, 1977.^ L. Greengard, The Rapid Evaluation of Potential Fields in Particle Systems, MIT, Cambridge, (1987).^ Rokhlin, Vladimir (1985). "Rapid Solution of Integral Equations of Classic Potential Theory." J. Computational Physics Vol. 60, pp. 187207.^ L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," J. Comput. Phys., 73 (1987), no. 2, pp. 325348.^ The Rubik's Cube Conjecture PROVEN! (Do we care?) Wednesday, September 08, 2010^ God's Number is 20.^ Math research team maps E8: Calculation on paper would cover Manhattan. MIT News. Elizabeth A. Thomson, News Office; March 18, 2007.^ E8 Media Blitz, Peter Woit.^ Mathematicians Map E8. Archived 2015-09-24 at the Wayback Machine By Armine Hareyan 2007-03-20 02:21.^ What is the way of packing oranges? Kepler's conjecture on the packing of spheres. Posted on May 26, 2015 by Antoine Nectoux. Klein Project Blog: Connecting mathematical worlds.^ Announcement of Completion. Flyspeck Project, Google Code.^ Proof confirmed of 400-year-old fruit-stacking problem. New Scientist, 12 August 2014.The Monte Carlo Method: Classic PapersMonte Carlo Landmark PapersRetrieved from " of computation. Computational method. Computational methodology.