

The Impact of Alan Turing: Formal Methods and Beyond

Jonathan P. Bowen^{1,2}[0000-0002-8748-6140]

¹ Southwest University
Centre for Research and Innovation in Software Engineering (RISE)
Faculty of Computer and Information Science
Chongqing 400715, China

² London South Bank University
School of Engineering
Borough Road, London SE1 1AA, UK

jonathan.bowen@lsbu.ac.uk
<http://www.jpbowen.com>

Abstract. In this paper, we discuss the influence and reputation of Alan Turing since his death in 1954, specifically in the field of formal methods, especially for program proving, but also in a much wider context. Although he received some recognition during his lifetime, this image was tarnished by the controversy at the time of his death. While he was known and appreciated in scientific circles, he did not enter the public's consciousness for several decades. A turning point was the definitive biography produced by Andrew Hodges in 1983 but, even then, the tide did not turn very rapidly. More recent events, such as the celebrations of his birth centenary in 2012 and the official British royal pardon in 2013, have raised Turing's fame and popularity among the informed general public in the United Kingdom and elsewhere. Cultural works in the arts featuring Turing have enhanced his profile still further. Thus, the paper discusses not only Turing's scientific impact, especially for formal methods, but in addition his historical, cultural, and even political significance. Turing's academic 'family tree' in terms of heritage and legacy is also covered.

1 Background

Alan Turing (1912–1954) has a rightful claim to the title of “Founder of Computer Science” [30]. He has also been called the “Father of Computer Science” [65]. Before World War II, Turing laid the theoretical groundwork for a universal machine that models a computer in its most general form. During the War, Turing was instrumental in developing and influencing actual computing devices that have been said to have shortened the War by up to two years by decoding encrypted enemy messages that were generally believed to be unbreakable [57]. Unlike some theorists he was willing to be involved with practical aspects and

was as happy to wield a soldering iron as he was to wrestle with a mathematical problem, normally from a unique angle.

With hindsight, Turing's 1936 seminal paper on computable numbers [154] foretold the capabilities of the modern computer. The War then brought about a radical, but perhaps fortuitous, change of direction in Turing's career, as his unique mathematical abilities were recognized during his time at Cambridge and he was invited to join Bletchley Park, the secret centre of the United Kingdom's efforts, to break German codes [116]. Decryption by hand was too laborious and time-consuming to succeed in the acutely limited time available. Turing recognized that machines, together with great human ingenuity, could tackle the problem far more quickly and reliably.

Despite his success in decryption, Turing was a victim of the times in which he happened to live. Homosexuality was illegal, remaining so in the United Kingdom until 1967. A mere fifteen years after Turing's arrest and prosecution, the atmosphere had become very different. But in the somewhat stifling and rigid Britain of the early 1950s, which was also the period of McCarthyism in America, a maverick and rather naïve person such as Turing was living in a dangerous world, unprotected by any public reputation. First and foremost, the secrecy surrounding his wartime work meant that public knowledge of his national importance was severely limited. Secondly, his ground-breaking 1936 paper would have seemed abstruse and not very relevant in a still largely pre-computer age to most scientists. Lastly, his later work on early artificial intelligence and morphogenesis made little practical impact in his lifetime, even among his closest colleagues.

1.1 Polymath

Alan Turing's interests were wide-ranging. This leads to the question of his field. Was Alan Turing a:

mathematician? He was certainly educated as a mathematician at Cambridge University and many of his papers were highly mathematical.

philosopher? His early ideas culminating in his 1952 paper on Artificial Intelligence (machine intelligence, as he called it) and the concept of the Turing test are admired by many philosophers.

computer scientist? His 1936 paper introducing the Turing machine, a simple generic model for an abstract computing device, is considered foundational for the discipline by many computer scientists. That said, the field was not formally recognized as an academic discipline until the 1960s [27].

codebreaker? His wartime work during World War II (1939–1945) on breaking the German Enigma code using the bombe machine is what he is best known for by many members of the general public.

biologist/chemist? His 1952 paper on morphogenesis has “chemical” in the title and is foundational for mathematical biology, a very active field currently with the concept of Turing patterns [43] in biological organisms and even elsewhere in nature.

The polymathic Turing is claimed by all these fields to a greater or lesser extent. He is known to have influenced people in many different fields, even in digital art using mathematical techniques inspired by his ideas on morphogenesis [29,39]. He is known to have been considering quantum effects towards the end of his life and perhaps physicists would claim him as well had he lived longer.

1.2 Genius

The term ‘genius’ is difficult to define precisely, but Alan Turing is generally acknowledged to be one. The 2010 book *Genius of Britain* accompanied a Channel 4 television series in the UK on scientists that changed the world [163] and Turing appears prominently in a chapter on *The Expediency of War*.

The mathematician Peter Hilton (1923–2010), a colleague of Alan Turing at Bletchley Park, wrote of Turing [90]:

... the experience of sharing the intellectual life of a genius is entirely different; one realizes that one is in the presence of an intelligence, a sensitivity of such profundity and originality that one is filled with wonder and excitement.

Andrew Robinson has posited that it normally takes around a decade to become a genius [136]. In Turing’s case, his first work of real genius was his 1936 paper [154], introducing what has become known as a Turing machine, an abstract version of a computer, at the age of 24. A decade earlier in the late 1920s, he was at Sherborne School, where at the age of 15 he handwrote a paper on Albert Einstein’s Theory of Relativity, demonstrating an understanding of Einstein’s ideas [153].

In a Lent Term 1927 Sherborne School report for Turing [145], at the age of 14, he was top of the form in mathematics with the comment:

Very good. He has considerable powers of reasoning and should do well if he can quicken up a little and improve his style.

Under Natural Science, it states “Chemistry. Good.” But Turing’s housemaster comments:

He is frankly not one who fits comfortably for himself into the ordinary life of the place – on the whole I think he is tidier.

At the bottom of the one-page report is a summary comment from Nowell Smith, in his last year as Headmaster of Sherborne:

He should do very well when he finds his *métier*: but meantime he would do much better if he would try to do his best as a member of this school – he should have more *esprit de corps*.

The following year in Michaelmas Term 1928, his young mathematics teacher, Donald Eperson, only eight years older than Turing, comments [25]:

He thinks very rapidly & is apt to be “brilliant”, but unsound in some of his work. He is seldom defeated by a problem, but his methods are often crude, cumbersome & untidy. But thoroughness & polish will no doubt come in time.

The new Headmaster, Charles L.F. Boughy, comments more optimistically than his predecessor that “This report is full of promise.” With hindsight, these comments are particularly apposite and some are farsighted, especially with respect to his powers of reasoning, finding his *métier*, increasing speed at mathematics, his brilliance, and promise, all indicative of Turing’s embryonic genius.

2 Scientific impact

Turing received an OBE (Officer of the Order of the British Empire) honour from the United Kingdom government for his codebreaking work at Bletchley Park in World War II and was made a Fellow of the Royal Society, the UK’s foremost scientific society, in 1951 [125]. However, the real recognition of his contribution came long after his death with the development of computer science [27] and as the truth of his crucial wartime role at Bletchley Park was gradually revealed. It is notable that Turing’s three most cited papers by far (by an order of magnitude compared to others, with more than 10,000 citations each in subsequent publications according to Google Scholar [78]) were published in 1936 [154], 1950 [157], and 1952 [159]. Each of these was foundational in the fields of theoretical computer science, artificial intelligence, and mathematical biology respectively. Without Turing’s premature death in 1954, soon after two of his three most influential publications, it is highly likely that he would have gone on to produce further inspirational ideas.

A lasting scientific memorial to Turing is the Association for Computing Machinery (ACM) *A.M. Turing Award* [4], presented to one and sometimes up to three leading computer scientists each year since 1966 by the world’s foremost computing association [42]. This is the highest scientific honour available to a computer scientist, widely considered as the equivalent to the Nobel Prize in the field. Many of the award winners have worked in areas for which Alan Turing was foundational, such as artificial intelligence and formal methods (the application of mathematics to software engineering).

33 Turing Award winners attended the *ACM A.M. Turing Centenary Celebration* on 15–16 June 2012 in San Francisco, USA, to honour Turing’s anniversary [2]. A.M. Turing Award winners are listed chronologically on the ACM website [5] and some who have worked in areas related to Turing’s interests are recorded in sections below. Although Turing died before the existence of computer science’s highest annual award, it is likely that most computer scientists would agree that he would have been very deserving of it [165].

2.1 Theoretical computer science

In theoretical computer science, the idea of a ‘Turing machine’ [66], a mathematical model of computation in the form of an abstract machine with an infinite

memory tape that can be read and written by the machine, has continued to be important in discussing what is and is not computable [83]. In addition, the related ‘halting problem’ on determining whether or not a given program terminates, building on the concept of the Turing machine, is a fundamental issue of reasoning about programs and a canonical problem in computer science [40,146]. It is a decision problem that has proved to be undecidable in the general case. Certainly Christopher Strachey (1916–1975), who knew Turing since they were both at King’s College, Cambridge, and was an expert programmer of both the Pilot ACE and Manchester Mark 1 computers, knew this as part of programming folklore even in the 1950s. He recounted how Turing gave him a verbal proof in a railway carriage in 1953 [128]. Strachey went on to found the Programming Research Group (PRG) at Oxford University, which became leading site for formal methods research.

The halting problem relates to program loops that may or may not continue forever. When proving programs correct, the concepts of partial and total correctness are important. The former only proves the program correct if it terminates whereas the latter also proves that the program does terminate, which requires additional effort for programs that include looping constructs. Hoare logic, devised by C.A.R. Hoare (born 1934, the 1980 A.M. Turing Award winner, who took over at the head of the PRG at Oxford after Christopher Strachey’s death), building on the ideas of Robert Floyd (1936–2001, the 1978 A.M. Turing Award winner) [75], provides a mathematical framework using an axiomatic approach that allows reasoning about the correctness of programs [93]. Turing himself published a very early proof of program correctness in 1949 (arguably the first) and this work was rediscovered and appreciated much later after his death [122]. Even computing developments since Turing’s death such as quantum computing [18] still rely on the concept of the Turing machine [67]. Turing machines have also influenced philosophy and the general concept of computability [59].

2.2 Artificial Intelligence

As well as theoretical computer science, Alan Turing can be considered a founding father for the field of Artificial Intelligence (AI) with his 1950 paper [157], although he himself used the term ‘machine intelligence’ and ‘AI’ was coined two years after Turing’s death, in 1956 by John McCarthy of Stanford University (1927–2011, the 1971 A.M. Turing Award winner) [84]. McCarthy emphasised the use of mathematical logic [115], leading to the Lisp functional programming language and the Prolog logic programming language, where programming is brought to the level of mathematical functions and relations, which surely would have been appreciated by Turing.

Marvin Minsky of MIT (1927–2016, the 1969 A.M. Turing Award winner) was also a leading light in the AI community and was influenced by Turing [89]. He discovered a four-symbol seven-state universal Turing machine in 1962. Donald Michie (1923–2007) [147], who himself worked at Bletchley Park contemporaneously with Alan Turing and Max Newman (1897–1984), Turing’s mentor at Cambridge in the 1930s [79]). He worked in the ‘Testery’ (a section named

after the founder, Major Ralph Tester) and discussed early ideas of problem solving using searching techniques with Turing at Bletchley Park, in the form of early AI [55]. Michie later established himself as a leading AI researcher at the University of Edinburgh, where the field remains a speciality to this day. He developed an early example of machine learning, based on the game of noughts and crosses (aka Tic-Tac-Toe), in the early 1960s and also founded the Turing Institute in Glasgow, which was active during 1983–94 as an applied AI research laboratory.

During the 1980s, there was a perceived threat from Japan due to significant funding in AI and the ‘Fifth Generation’ of computing [74]. This helped to prompt the funding of the UK Alvey Programme for Advanced Information Technology that supported collaborative projects between academia and industry. AI continues to be of wide interest, with attempts to perform the Turing test successfully [12] and interest in the idea of ‘superintelligence’ where machines can outperform humans in their general intelligence [24].

Developments in deep learning, based on machine learning using learning data representations rather than task-specific algorithms (e.g., learning to play games in general rather than just a specific game like chess for example) using approaches such as neural networks are now proving to be increasingly possible in practice [63]. Deep learning is an area that most likely would be of interest to Turing and to which he could have contributed his ideas.

2.3 Mathematical biology

Turing’s 1952 paper on morphogenesis [159] was foundational in the field of mathematical biology; it demonstrated that relatively simple mathematics can generate seemingly complex biological patterns and shapes through chemical reactions. In fact, according to Google Scholar [78], this is Turing’s most cited paper, with over 600 citations annually in recent years. The approach has become increasingly important in the understanding of biological processes. It is interesting to note that Turing’s original paper cited only two references, whereas nowadays it is normal to include far more citations, including foundational papers, such as those produced by Turing. This was not uncommon for innovative papers in the past; for example, Albert Einstein included no citations in his 1905 paper on relativity.

2.4 General computer science

The Hungarian-American mathematician John von Neumann (1903–1957) was another great foundational figure of modern computing. The term ‘von Neumann architecture’ is used to describe the standard architecture of a computer, as described by him in 1945. This is often compared with a Turing machine, but the two serve very different purposes. A Turing machine is a theoretical model to aid in reasoning about computation, whereas the von Neumann architecture is a more practical description of the configuration of a standard electronic digital

computer. Von Neumann knew of and acknowledged Turing's pioneering work. A colleague of von Neumann at Los Alamos, Stan Frankel, noted [134]:

I know that in or about 1943 or '44 von Neumann was well aware of the fundamental importance of Turing's paper of 1936 . . . Von Neumann introduced me to that paper and at his urging I studied 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 sure that he would never have made that mistake himself. He might well be called the midwife, perhaps, but he firmly emphasized to me, and to others I am sure, that the fundamental conception is owing to Turing – in so far as not anticipated by Babbage, Lovelace, and others.

So, the main contender to Turing's impact with respect to the foundations of computing acknowledged the leading role of Turing and the scientific debt due to him.

Turing was a mathematician and philosopher at a time when computer science did not exist as a separate discipline [27]. As well as being foundational for a new discipline, Turing has also been highly influential in mathematics and philosophy. Perhaps a leading example is the provocative 1989 book *The Emperor's New Mind* by the Oxford mathematician and philosopher of science Roger Penrose (born 1931) [130]. The first two chapters on 'Can a computer have a mind?' and 'Algorithms and Turing machines' are largely based around Turing's 1936 and 1950 papers [154,157]. The book goes on to cover the philosophical 'mind-body problem', attempting to explain how mental states, events, and processes are related to physical manifestations of these. Penrose's subsequent 1994 book *Shadows of the Mind* [131] later argues that human consciousness is not algorithmic and thus cannot be modelled by a Turing machine. He posits that quantum effects may be a critical part of consciousness and that the human mind has qualities that no Turing machine can possess. With Turing's interest in quantum mechanics, he could well have contributed to Penrose's debate.

Subsequent models of computation have all been related to Turing machines in some way. John Conway's 'Game of Life' introduced in 1970 is an example of a two-dimensional space forming a cellular automaton with very simple rules that has the potential to produce complex patterns and undertake computation, with the power of a universal Turing machine. It can even be used to model a Turing machine visually [29,135]. Stephen Wolfram has studied cellular automata extensively, as covered in his 2002 book *A New Kind of Science* [169]. This includes much material on Turing machines in particular, as well as other ideas conceived by Turing.

The two-volume set *The Legacy of Turing* of 1996 covers aspects of artificial intelligence and computer science in Volume I, together with philosophy and cognitive science in Volume II, with an extensive set of contributors [119]. In 2004, some of Turing's most important scientific contributions, with annotations, were compiled and annotated by Jack Copeland in *The Essential Turing* [54]. An archive of Turing-related material by the same editor is also available online [60].

The centenary of Turing's birth in 2012 led to a number of special issues and articles in scientific journals themed around the achievements of Turing. The leading scientific journal in the world, *Nature*, featured Turing on the front cover during 2012 for a special issue entitled *Alan Turing at 100* [123]. One of the foremost professional magazines in computing, the *Communications of the ACM*, also had a special article on Turing in 2012 and featured a slate sculpture by Stephen Kettle of Turing at Bletchley Park [108] on the front cover [50]. The year of 2012 was obviously a highpoint for interest in Turing, scientific and otherwise, but still his importance to computer science, mathematics and philosophy remains undiminished.

2.5 Formal methods and program proving

The application of mathematics to the specification and development of computer-based systems, especially software, but also hardware, has been dubbed 'formal methods' [36] since around the late 1970s. The term has been borrowed from the field of mathematical logic (e.g., see the 1962 book on logic by Evert Willem Beth entitled *Formal Methods* [20]). Beth used the term even earlier in his 1955 book *Semantic Entailment and Formal Derivability* [19].

Much earlier in 1910, Paul Carus wrote in conclusion on the Saxon term 'kenlore', describing the process of cognition, in the context of epistemology [47]:

From this norm which dominates the world and which is reconstructed in our mind we derive those principles of all our purely *formal methods*, our principles of logic and logical necessity, of universality, of our fundamental conditions for mathematical thought and geometrical constructions, and here accordingly lies the corner stone of kenlore.

The term was also used by Bertrand Russell in the context of logic in 1912 [142].

Alan Turing wrote what can be considered the first 'formal methods' paper in the context of program proving in 1949, in a short (three-page) paper entitled *Checking a Large Routine for a Conference on High Speed Automatic Calculating Machines* at Cambridge [156]. This has subsequently been examined in detail in 1984 [122]. Although it contains minor errors in the detail, not atypical for Turing, the overall idea presented was correct. The following are some short extracts from Turing's 1949 paper [156] with some relevant words emphasised;

- "In order to assist the checker, the programmer should make *assertions* about the various states that the machine can reach."
- "The checker has to *verify* that the ...initial condition and the stopped condition agree with the claims that are made for the routine as a whole."
- "He has also to *verify* that each of the assertions ... is correct."
- "Finally the checker has to *verify* that the process comes to an end."

None of these comments would be out of place in a formal methods paper of today, except that the "checker" would most likely now be computer-based software (perhaps with human guidance) rather than a human alone. The paper also states:

The following convention is used:

- (i) a dashed letter indicates the value at the end of the process represented by the box;
- (ii) an undashed letter represents the initial value of a quantity.

This could be describing an operation schema box in the modern-day Z notation [28]. Further work on Turing's 1949 paper and its impact has been undertaken more recently by Cliff Jones [104,105,106].

The Dutch mathematician and early computer scientist Aad van Wijngaarden (1916–1987), a pioneer of programming language grammars (after whom the term ‘Wijngaarden grammar’ was coined for his two-level approach to grammars) was at the 1949 Cambridge meeting where Turing presented his pioneering paper on program proving [156], but it had no great influence on his research at the time [105]. Only by the 1960s is there evidence of influence [104]. Peter Naur (1928–2016, the 2005 ACM A.M. Turing Award winner), best known for BNF notation (Backus-Naur Form) with John Backus (1924–2007) for the formal description of programming language syntax, considered proofs of algorithms in 1966 [124].

Robert W. (Bob) Floyd (1936–2001) and others rediscovered ideas for program proving similar to those of Turing in the 1960s [75]. C.A.R. (Tony) Hoare developed these further with his axiomatic approach based on assertions [93]. Had Turing lived longer, perhaps formal methods (in particular, program proving) would have developed more rapidly, rather than being rediscovered more than a decade later. As with most of Turing's research interests, he was well ahead of his time compared with others because of his novel approach to problems, working from first principles.

Turing's 1949 paper was essentially ignored during the 1950s. However, by the 1960s, Turing's ideas on program proving were being rediscovered independently. Figure 1 shows some key publications in formal methods with respect to proving programs correct from Turing's 1949 paper onwards during the second half of the 20th century [122].

Publications in the 1960s relating to formal methods were mainly foundational research papers. Later key publications were mostly works leading on from these papers in the form of tutorial-style books. In the 1970s, structured programming became prominent, as extolled by Ole-Johan Dahl (1931–2002), Edsger W. Dijkstra (1930–2002), and Tony Hoare [61] (all ACM A.M. Turing Award winners in 2001, 1972, and 1980 respectively). *The Science of Programming* book [81] by David Gries (born 1939) builds on Dijkstra's book *A Discipline of Programming* [68], which concentrates on abstraction, suggesting that a program and its proof of correctness should be developed in tandem. The 1986 book *Systematic Software Development Using VDM* by Cliff Jones (born 1944) provides an approach to refinement from a formal specification towards a program in the context of VDM (Vienna Development Method), an early formal method.

The paper *Laws of Programming* presents some general algebraic laws for imperative programming languages [94]. The work of Carroll Morgan (born 1952)

1949: Alan Turing, <i>Checking a Large Routine</i> [156].
1966: Peter Naur, <i>Proof of Algorithms by General Snapshots</i> [124].
1967: Robert Floyd, <i>Assigning Meaning to Programs</i> [75].
1969: Tony Hoare, <i>An Axiomatic Basis for Computer Programming</i> [93].
1972: Ole-Johan Dahl et al., <i>Structured Programming</i> [61].
1976: Edsger Dijkstra, <i>A Discipline of Programming</i> [68].
1981: David Gries, <i>The Science of Programming</i> [81].
1986: Cliff Jones, <i>Systematic Software Development Using VDM</i> [103].
1987: Tony Hoare et al., <i>Laws of Programming</i> [94].
1990: Carroll Morgan, <i>Programming from Specifications</i> [121].
1996: Jean-Raymond Abrial, <i>The B-Book</i> [1].
1998: Tony Hoare & He Jifeng, <i>Unifying Theories of Programming</i> [95].

Fig. 1. Some key 20th-century publications on program proving.

on *Programming from Specifications* [121] explicitly depends on [68,75,93], as stated in the book's preface. Jean-Raymond Abrial (born 1938), progenitor of the Z notation [28], later produced *The B-Book: Assigning Programs to Meanings* [1] on the B-Method to derive a program from a formal specification in a rigorous manner with tool support.

The book on *Unifying Theories of Programming* (UTP) [95] by Hoare and the Chinese computer scientist He Jifeng (born 1943) aims to provide a coherent formal basis for the specification, design, and implementation of programs, using denotational, algebraic, and operational styles of semantics in a unified framework. It has spawned an international research community with its own regular UTP conference that is still active after 20 years of the existence of UTP. The bibliography in the book cites all the previous publications listed in Figure 1, including the 1949 Turing paper [156].

Turing made significant contributions to mathematics, philosophy, computer science, and even mathematical biology. His multifarious interests extended to what is now known as formal methods, especially with respect to proving programs correct. Developments have continued in proving computer systems at various levels of abstraction [91], yet there are still significant issues, such as scaling to handle large systems.

Despite their benefits, not all have been convinced that formal methods are worthwhile in practice [35]. However, formal methods communities continue [31,38], especially for application in high-integrity systems where safety or security is important [39]. The state of the art moves on [22] and increasingly complex systems can be tackled using formal methods [92]. Most comprehensive books on software engineering include a section on formal methods [152].

3 Academic legacy and heritage

3.1 Family scientific heritage

The Turing family itself, apart from Alan Turing, has been undistinguished scientifically, although there is a Turing Baronetcy, created in 1638. The 12th Baronet, Sir John Dermot Turing (born 1961), a solicitor, is a nephew of Alan Turing and has written a 2015 biography about him [161]. However, Alan Turing's mother was Ethel Sara Turing (née Stoney, 1881–1976), who herself wrote a biography of her son, originally published in 1959 and republished for Turing's centenary in 2012 [162]. Her father was Edward Waller Stoney (1844–1931), chief engineer of the Madras Railways in India.

The Anglo-Irish Stoney family included George Johnstone Stoney FRS (1826–1911), an Irish physicist most famous for suggesting the term 'electron' (initially 'electrine') [161], highly apt with respect to Turing's early pioneering work on electronic computers. His brother was Bindon Blood Stoney FRS (1828–1909), an engineer and astronomer. His children included George Gerald Stoney FRS (1863–1942), a mechanical engineer, Edith Anne Stoney (1869–1938), a medical physicist, and Florence Ada Stoney OBE (1870–1932), a radiologist. The Stoney family genes meant that Turing's family heritage included both scientists and engineers, an excellent grounding for Turing's combination of theoretical and practical genius.

3.2 Academic background

Academically, after studying mathematics at King's College, Cambridge, as an undergraduate, Turing was then supervised by the mathematician and logician Alonzo Church (1903–1995) for his PhD studies at Princeton University in the United States, which he completed in 1938 [155] (see Figure 2) with less than two years of study [133]. Alan Turing himself only supervised two PhD students, both at the University of Cambridge, according to the Mathematics Genealogy Project [118]. His most famous student was the mathematician and logician Robin Gandy (1919–1995), who subsequently moved to the University of Oxford.

Dana Scott (born 1932, the joint 1976 A.M. Turing Award winner with Michael O. Rabin) was a later PhD student of Alonzo Church at Princeton who was also based at Oxford from 1972 to 1981. He worked with Christopher Strachey (1916–1975), a former colleague of Turing at the University of Manchester, on the denotational semantics of programming languages, which became dubbed the Scott-Strachey approach [148]. One of Dana Scott's students at Oxford was the philosopher Jack Copeland (born 1950), subsequently a leading Turing scholar [54,56,58,59,60], based at the University of Canterbury in New Zealand. He was also a colleague of Robin Gandy at Oxford as indicated by the additional horizontal arrow link in Figure 2.

As well as Robin Gandy, Turing co-supervised Beatrice Worsley (1921–1972) at Cambridge, with the mathematician and physicist Douglas Hartree (1897–1958) in the EDSAC group of Maurice Wilkes (1913–2010, the 1967 ACM A.M.

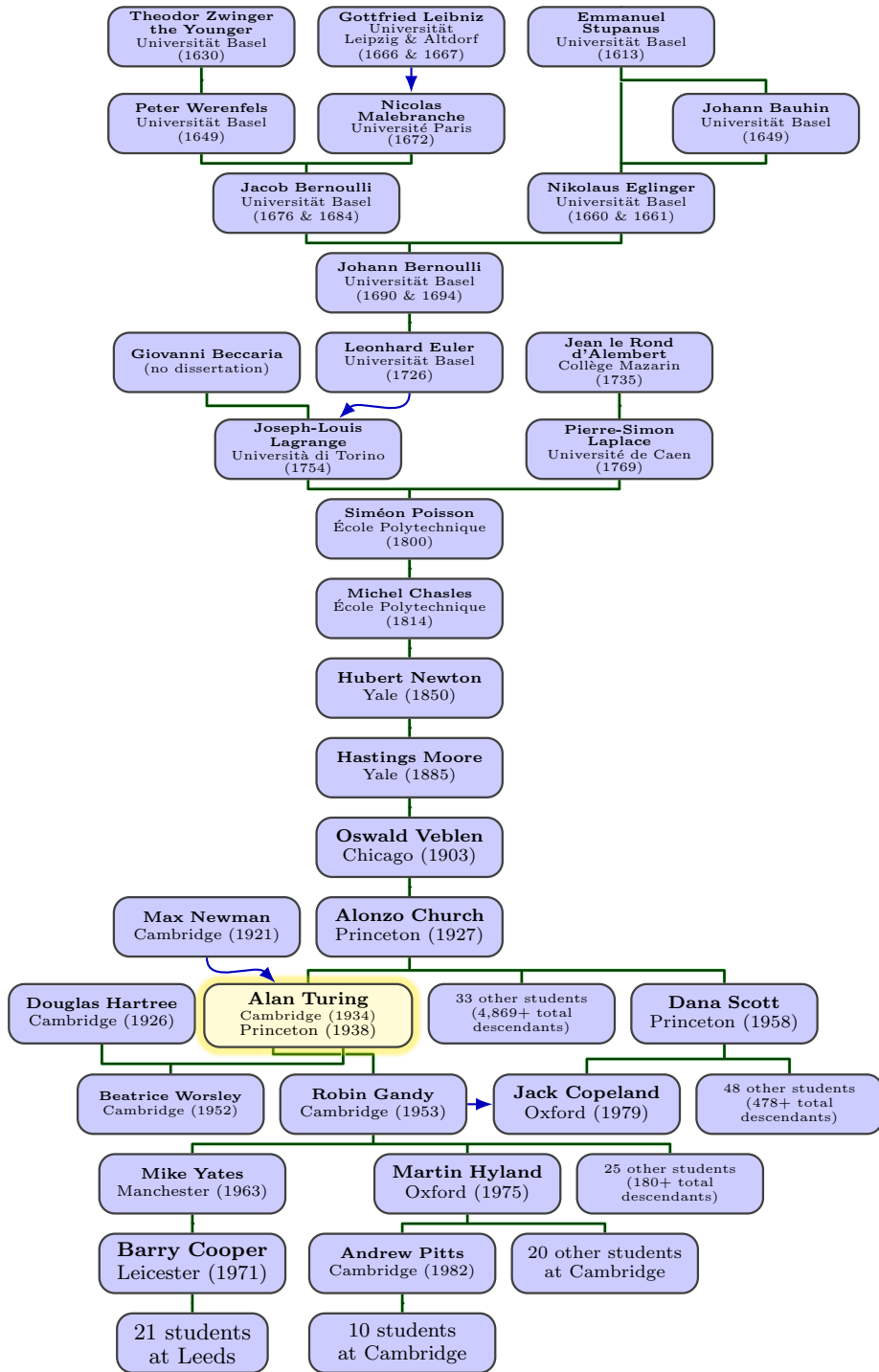


Fig. 2. Academic supervisor tree for Alan Turing [118].

Turing Award winner). This was arguably the first computer science doctorate, entitled *Serial Programming for Real and Idealised Digital Calculating Machines* and submitted in May 1952 [170]. Worsley helped to write the first program to run on the EDSAC computer at Cambridge. She was subsequently a very early computer science academic based in Canada before her premature death in 1972 at only 50 years old [44]. Douglas Hartree was himself supervised by the eminent New Zealand physicist Ernest Rutherford (1871–1937, discoverer of the proton and known as the father of nuclear physics), who in turn was supervised by Joseph Thomson (1856–1940, discoverer of the electron), both at Cambridge and both Nobel prize winners, so Beatrice Worsley had an outstanding supervision pedigree.

Robin Gandy supervised 27 students, three at the University of Manchester and the rest at Oxford. Of these, the mathematical logician and theoretical computer scientist Martin Hyland had 21 doctoral students, all at Cambridge, including the theoretical computer scientist Andrew Pitts, also based at Cambridge. Another student, at Manchester, was Mike Yates, who himself supervised the mathematician and computability theorist Barry Cooper (1943–2015) at Leicester, later Professor of Pure Mathematics at the University of Leeds. Barry Cooper was a major instigator of the 2012 centenary celebrations for Alan Turing in the United Kingdom, including at Manchester [51], resulting in a later publication [52].

3.3 Mathematics Genealogy Project

The ancestry for Alan Turing’s line of supervisors can be traced back on the Mathematics Genealogy Project website [118]. Turing’s supervisor Alonzo Church was supervised by the mathematician, geometer, and topologist, Oswald Veblen (1880–1960), at Chicago. It has been claimed that Veblen, rather than Church or Turing for example, was responsible for the start of computing at Princeton University [71]. Veblen taught mathematics at Princeton between 1905 and 1932. He was one of the two original faculty members in the Institute for Advanced Study (IAS), along with Albert Einstein (1879–1955).

In 1930, Veblen invited John von Neumann (1903–1957), then aged 27, to join Princeton as a lecturer in quantum statistics. Von Neumann joined the IAS itself in 1933 and was of course highly influential in early computing with his von Neumann architecture for a computer, a more practical model than the theoretically oriented Turing machine. Sharing an interest in topology, Veblen was also helpful in encouraging Max Newman to visit Princeton as early as 1928 [166]. This in turn led to Newman encouraging Turing, while both at Cambridge, to study for his PhD at Princeton under Church. So, Veblen was indirectly responsible for Turing’s presence at Princeton, as well as von Neumann.

Newman spent six months at Princeton in 1937 when Turing was working on his PhD there. However, later during World War II, Veblen opposed Newman visiting Princeton due to wartime considerations [126]. This was fortuitous for Britain’s wartime effort since although Newman was initially reticent, due to his German background on his father’s side (despite being of Jewish origin), he

joined Bletchley Park in 1942, where Turing was already based. There he led the ‘Newmanry’, which worked on breaking the German Lorenz cipher, for which the Colossus computers were developed [53].

3.4 Historic academic lineage

Turing’s academic pedigree can be traced back more historically, as covered in this subsection for the interested reader.

Veblen was supervised by the mathematician Eliakim ‘Hastings’ Moore (1862–1932) at the University of Chicago. Moore was supervised by the astronomer and mathematician Hubert Anson Newton (1830–1896), at Yale University. Newton, also at Yale, studied under the French mathematician Michel Floréal Chasles (1793–1880).

Then the line moves from the United States to France with Michel Chasles who studied under the mathematician, engineer, and physicist, Siméon Denis Poisson (1781–1840) at the École Polytechnique in Paris. Poisson is known for Poisson’s equation, a partial differential equation that is useful in mechanical engineering and theoretical physics, and the Poisson distribution in probability theory and statistics, among other contributions to mathematics. Poisson studied under the Italian-Frenchman Joseph-Louis Lagrange (1736–1813) and Pierre-Simon Laplace (1749–1827) at the École Polytechnique, both very eminent mathematicians, and both also with an interest in astronomy. Lagrange is known for the reformulation of classical (Newtonian) mechanics, Lagrangian mechanics, using Lagrange equations, among other mathematical contributions. Laplace is especially known for the Laplace transform, used in the transformation of integrals.

Laplace’s adviser was Jean-Baptiste le Rond d’Alembert (1717–1783), the polymathic French mathematician, mechanician, philosopher, music theorist, and physicist. According to the Mathematics Genealogy Project, d’Alembert’s adviser is unknown so the lineage cannot be traced back further. Lagrange’s adviser was the Italian physicist Giovanni Battista (Giambattista) Beccaria (1716–1781) in Turin. Beccaria’s adviser is unknown, and no dissertation is known. However, Laplace also worked with and was highly influenced by the Swiss astronomer, engineer, logician, mathematician, and physicist, Leonhard Euler (1707–1783), largely through correspondence by letter. Laplace wrote: “Read Euler, read Euler, he is the master of us all” [69].

Euler undertook his doctorate with the Swiss mathematician Johann Bernoulli (1667–1748) as his advisor at Basel. Johann Bernoulli undertook two dissertations, one under Jacob Bernoulli (1655–1705) and another under Nikolaus Eglinger (1645–1711), both at Basel. Jacob Bernoulli studied under the Swiss theologian Peter Werenfels (1627–1703) and the French priest and rationalist philosopher Nicolas Malebranche (1638–1715) for two dissertations, both at Basel. Peter Werenfels studied under the preacher and theology professor, Theodor Zwinger the Younger (1597–1654) at Basel. Nicolas Malebranche met the renowned German polymath, mathematician and philosopher, Gottfried Wilhelm Leibniz (1646–1716) in Paris and they corresponded after this. They dis-

cussed the laws of motion extensively. Leibniz conceived differential and integral calculus, independently of Isaac Newton (1642–1726/27). Leibniz’s notation for calculus has become the accepted notation by mathematicians.

Nikolaus Eglinger studied under the Swiss physician Emmanuel Stupanus (1587–1664, aka Stuppan) and the Swiss botanist and physician Johann Bauhin (1541–1613) for two dissertations, both at Basel. Bauhin himself also studied under Stupenus at Basel.

Thus, Turing’s academic lineage goes back to Euler and Leibniz, two of the most renowned mathematicians in history, and beyond, as well as Lagrange, Laplace, and two members of the Bernoulli family. Turing’s supervisor lineage, both historically and for those after him, is illustrated in Figure 2.

3.5 Turing’s mentor

Although not officially his supervisor, Max Newman was a great mentor for Turing [79], both early on at Cambridge and later at Manchester, as well as at Bletchley Park during World War II. Newman’s 1935 lectures on the foundations and mathematics at Cambridge inspired Turing to write his 1936 paper introducing the concept of Turing machines [154], which Newman recognized as a novel and important piece of work. Without Newman, it could well have not been published and Turing’s genius might not have been recognized so quickly. Newman himself never studied for a doctorate.

Newman was a ‘Wrangler’ (gaining a first-class undergraduate degree in mathematics) at St John’s College, Cambridge, in 1921. He wrote a thesis on using ‘symbolic machines’ to make physical predictions, a precursor to his interest in computing later in his career [126]. Newman spent 1922–3 at the University of Vienna [79], but returned to Cambridge in 1923 with a Fellowship at St John’s College. He became a lecturer in mathematics for Cambridge University in 1927 and his 1935 lectures were attended by the young Turing. Later it was Newman who encouraged Turing to join Manchester University after World War II, working on early computers there.

4 Turing eponyms

Alan Turing’s name is associated with a number of different concepts and ideas related to his work. Many of these are notable enough to have their own individual entries on Wikipedia [167].

‘Turing machine’ and ‘Turing test’ are well-known Turing terms resulting from the ideas in Turing’s 1936 and 1950 papers [154,157], with many specialized versions of both [167]. Increasingly in use is the term ‘Turing pattern’, often used in the plural, to describe the complex patterns generated by mathematics based on that presented in his 1952 paper on morphogenesis [159] (e.g., see [43]).

The previously mentioned ‘Turing Institute’ (1983–94) was an AI laboratory in Glasgow, Scotland, established by Donald Michie, a wartime colleague

of Turing. The newer and unrelated ‘Alan Turing Institute’ is a national governmental data sciences institute in the United Kingdom, founded in 2015, located in London, and overseen by a group of leading UK universities.

There are a number of roads, streets, and buildings named after Turing in the UK and elsewhere, especially in locations associated with him. Examples include Alan Turing Buildings in Guildford and Manchester, Alan Turing Roads in Guildford and Loughborough, and Alan Turing Way in Manchester. There are a Turing Drive in Bracknell, Turing Gate near Bletchley Park, and Turing Roads in Biggleswade and Teddington, west London, the latter very near the National Physical Laboratory where Turing worked after World War II.

The BCS (British Computer Society), the professional IT association in the UK, holds an annual Turing Lecture jointly with the IET (Institution of Engineering and Technology). Finally, and most internationally, as mentioned earlier in this paper, Turing is honoured by the annual ‘ACM A.M. Turing Award’, the world’s highest award in computer science, and a fitting tribute to Turing for providing the foundations of the discipline.

More recently, a 2017 law in England and Wales that pardons gay people for what were historically offences has been dubbed the ‘Alan Turing law’, also just ‘Turing law’ or ‘Turing’s law’ [14,120]. This will be discussed in more detail later in the paper. Further Turing terms are presented in [30] and many more eponyms can be found on Wikipedia citewik18 for the interested reader.

5 Historical impact

Turing is considered an important figure in the overall history of science. For example, the 1991 book *A History of Knowledge* attempts to cover the entire range of human invention and creativity, and yet devotes a section to Alan Turing and Turing machines even in such an all-encompassing book [164]. *The Oxford Companion to the History of Modern Science* (2003) naturally includes an entry for ‘Computer Science’ [27], but also a separate entry for ‘Artificial Intelligence’, and both cover Turing’s foundational role [85]. In addition, there is a specific entry for Turing himself, which notes presciently that “Turing’s status as a cult hero will undoubtedly increase.”

Any book covering the history of computing would be incomplete without an explanation of Turing’s role in it. The 1979 book *The Mighty Micro* on the impact of the computer revolution by the British computer scientist and psychologist Christopher Evans (1931–1979) has many entries in the index for Alan Turing and the Turing test but interestingly not specifically for the Turing machine [73]. There is a concentration on whether machines can think, with Evans’ psychological background, including reference to Turing’s 1950 paper on machine intelligence [157]. Simon Lavington’s 1980 book on *Early British Computers* naturally has multiple entries on Turing. It includes a chapter on the development of Turing’s ACE computer and a photograph of the eventually constructed Pilot ACE on the back cover [110].

The detailed biography *Alan Turing: The Enigma* by Andrew Hodges dating from 1983 and reissued in 2012 for Turing’s centenary has played a significant part in increasing Turing’s visibility over the years since it originally appeared, when the secrecy surrounding the wartime Bletchley Park had only relatively recently started to be lifted in the 1970s [96]. It remains the definitive account of Turing’s life, from both professional and personal viewpoints, with which other biographies of Turing have difficulty in competing [149]. Hodges also produced an early website on Turing associated with his book in the early 1990s [98], linked with the online *Virtual Museum of Computing*, and still in existence today [32].

Figure 3 shows a graph of mentions of Alan Turing in books digitized on Google Books, using the online Ngram Viewer facility, which has a database of phrases mentioned in books by year (apart from the most recent decade). Since the late 1970s, after the significance of Bletchley Park began to be revealed, there has been a steady rise in mentions of Alan Turing in books over the years. There was a very significant peak in 1983, the original year of publication of Andrew Hodges biography of Turing [96]. It will be interesting to see the peak that is likely around the 2012 centenary of Turing when this is available.



Fig. 3. Graph of mentions of ‘Alan Turing’ in books (1960–2005). (Ngram Viewer, Google Books: <http://books.google.com/ngrams>.)

The practically oriented and wide-ranging book *Bit by Bit* on the history of computers by the American author Stan Augarten in 1984 has a significant section on Alan Turing and Turing machines [8]. *The Silicon Idol*, also published in 1984, covered the microprocessor revolution and its social impact [143]. Despite its relatively modern focus on the recent period before its publication, Alan Turing features a number of times in sections on thinking machines and the origins of computing.

The 1991 book *The Dream Machine* was associated with a BBC television series in the United Kingdom covering the computer age, including multiple references to Alan Turing [129]. In 1997, *Darwin Among the Machines* covered the emergence of thinking machines, in which Turing played a key early part

[70]. All of Turing's most significant achievements are included in the book. *The Code Book* by Simon Singh in 1999 surveyed the history of secret messages from the days of Ancient Egypt to the use of quantum cryptography [144]. Turing is mentioned extensively along the way, mainly with respect to the breaking of the Enigma code using the Bombe machines.

In 1999, *Time* magazine published *The Great Minds of the Century* (Time 100) [151], covering the 20th century, including Alan Turing among its selection, labelled as a 'computer scientist' [80]. The mathematician Kurt Gödel, whose ideas influenced Turing (introduced to him through Max Newman's lectures in Cambridge), and the philosopher Ludwig Wittgenstein, whose lectures Turing attended at Cambridge, were also listed, together with John von Neumann, who acknowledged and appreciated Turing's contribution to computing ideas. The only more recent computer scientist (trained as a physicist) to be included was Tim Berners-Lee (born 1955), inventor of the World Wide Web.

Turing's impact is also felt in all the institutions for which he worked. The National Physical Laboratory (NPL), where the Pilot ACE computer was produced based on Turing's ACE designs, continued pioneering work, most notably with computer communications and the development of packet-switching in the 1960s, led by Donald W. Davies (1924–2000), a colleague of Turing [171]. King's College, Cambridge, where Turing studied as a student and was later a Fellow, holds an extensive archive of Turing-related material, some of which is available online in the *Turing Digital Archive* [109]. The 2007 Alan Turing Building at the University of Manchester houses the School of Mathematics there.

Some accounts of Turing and his contemporaries, with regard to building early computers, appeared for the 2012 centenary of Turing's birth in a collection of contributions edited by the British historian of computing Simon Lavington [111]. The collection includes a section covering the legacy of Turing in the context of early British computers [112].

The 2012 *Alan Turing Year* [51], led by the mathematician and computability theorist Barry Cooper (1943–2015), with the help of others, coordinated a significant number of celebratory events, especially at institutions associated with Turing and during the month of June. Many of the authors in the volume were associated with three of these meetings on successive weekends in June 2012: *ACE 2012 (Alan Turing's 100th Birthday Party)* was held at King's College, Cambridge, Turing's college; *Turing's Worlds* took place at Rewley House, Oxford, in association with the British Society for the History of Mathematics (BSHM); and the *Turing Educational Day* (TED) was held at Bletchley Park. A major event at the University of Manchester with over 70 leading experts led to an associated book, *Alan Turing: His Work and Impact*, providing in-depth coverage of Turing's scientific influence [52].

More recent accounts of the history of computers [45] and computer science [62] continue to devote significant space to Turing's ideas. Even though computer science can be considered to have started with Charles Babbage, there was a significant reawakening of ideas in modern computing through the work of Turing that continues to be important today, especially at the theoretical level.

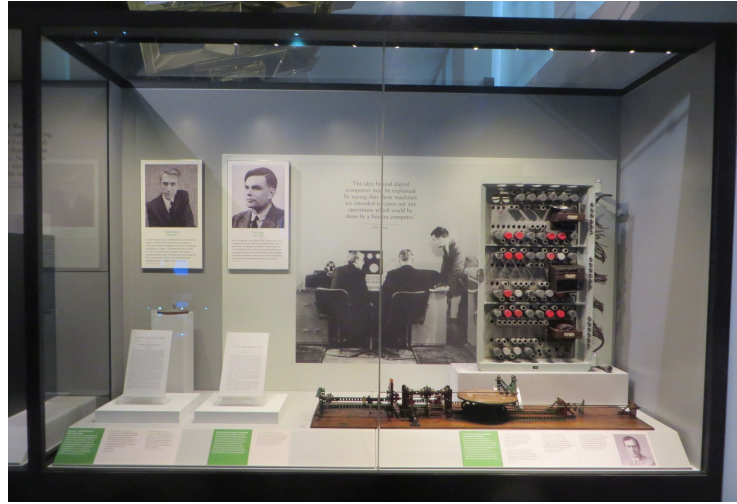


Fig. 4. An exhibit on Alan Turing and Claude Shannon, in the Information Age gallery [21] at the Science Museum, London. (Photograph by Jonathan Bowen.)

The London Science Museum has the original Pilot ACE computer, developed at the National Physical Laboratory (NPL) from Turing’s more ambitious ACE design, on permanent display in the Information Age gallery [21]. This gallery includes a showcase (see Figure 4) featuring both Alan Turing and Claude Shannon (1916–2001), considered by many to be the “father of information theory”. The showcase includes the following apt quotation:

“The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer.” – Alan Turing (see Figure 4)

Turing and Shannon met during World War II at Bell Labs in New York [77] and discussed ideas such as machine intelligence. Turing also features in the recently updated mathematics gallery at the museum, especially with respect to his work on breaking the German Enigma code at Bletchley Park during World War II [141]. Turing himself visited the Science Museum in 1951 and was fascinated by an electromechanical cybernetic ‘tortoise’ (a small autonomous robot) on display that could detect and respond to its surroundings [88,102].

During 2012–13 for the Turing centenary celebrations, a special exhibition, *Codebreaker – Alan Turing’s Life and Legacy* [140], was held at the Science Museum, featuring the Pilot ACE, Enigma machines, and other Turing-related items, including the ‘tortoise’ that he viewed in 1951 [87]. The Pilot ACE was used in calculations concerning the fatal flaw in the rectangular windows of the early Comet passenger jet aircraft of the 1950s. The exhibition covered such

historical impact and overall was a fitting celebration of Turing's life and achievements.



Fig. 5. Slate sculpture of Alan Turing with an Enigma machine at Bletchley Park, by Stephen Kettle [29,108]. (Photograph by Jonathan Bowen.)

6 Popular and cultural impact

Turing's life, cut short as it was at the age of 41, has attracted works from the arts, including the 1986 play *Breaking the Code* by Hugh Whitmore, the 1995 novel *Enigma* by Robert Harris, *A Man from the Future* in 2014 by the Pet Shop Boys, and sculptures in Manchester and at Bletchley Park (see Figure 5), to name but a few examples [30].

Alan Turing's life was dramatized in the 2014 film *The Imitation Game*, especially with respect to his codebreaking work during World War II [99]. Since then, the 2015 biographical film *Steve Jobs* directed by Danny Boyle featured Turing in background pictures [49]. The bitten apple used by Steve Jobs as his Apple company logo has been linked with Turing, even if erroneously [76,100]. The film alludes to this in a passage in which Steve Jobs admits that it is not so, but wishes that it were true [138].

6.1 Memorabilia

Items associated with Alan Turing have become increasingly valuable, especially his papers and notes. Google helped Bletchley Park acquire preprints of papers by Turing owned by Max Newman, some annotated by Turing, for a significant sum in 2011 and these are now on display there [117]. In 2015, a 56-page notebook by Alan Turing, subsequently owned by his student and close friend Robin Gandy, sold at auction in New York for more than a million dollars [86]. This is the only known extensive handwritten manuscript by Turing, which he authored at Bletchley Park in 1942.

In May 2017, a filing cabinet at the University of Manchester was found to contain around 150 documents relating to Turing, dating from 1949 until his death in 1954 [15]. They provide some insight into his personal views as well as his professional life at Manchester. One briefly documents a debate on whether machines can think, opened by Max Newman following by Turing, with Richard Braithwaite and Douglas Hartree also contributing [114]. R.B. Braithwaite (1900–1990), a Fellow at King’s College, Cambridge, was a philosopher with an interest in science and religion.

A similar debate was recorded by the BBC on 10 January 1952 for broadcast twice on the radio later that month [160]. This was opened and led by Braithwaite. Instead of Hartree, Sir Geoffrey Jefferson (1886–1961), a neurologist and pioneering neurosurgeon based at the University of Manchester, contributed to this discussion. Although the original recording has been lost, a transcript has survived [160]. Turing’s ideas on whether machines can think raised much debate in the early 1950s, although not all were convinced. For example, see an article by Maurice Wilkes [168], who was not completely enamoured of Turing and his ideas.

Turing also delivered a 1951 BBC radio broadcast on thinking machines, in a series with other scientists [105,102], entitled *Can Digital Computers Think?*. Turing’s original typewritten transcript still exists [158]. The broadcast was heard by Christopher Strachey, who was prompted to write to Turing on the subject [7]. Despite the fact that Turing was broadcast on the radio, it is unfortunate that there is no known audio or film recording of him, or even a colour photograph, in existence.

In November 2017, a handwritten and autographed two-page letter by Alan Turing was sold for UK £75,000 at Bonhams in London [23]. The letter was written by Turing from his home in Wilmslow around 1950 (although the letter is undated) to his former mathematics teacher at Sherborne School, Donald Eperson (1904–2001) [127], who had by then become vicar at the church in the village of Charminster, Dorset, in southern England not far from Sherborne. They shared an interest in the puzzles and logic of Charles Lutwidge Dodgson (1832–1898), aka Lewis Carroll, author of the ‘Alice in Wonderland’ books. Eperson studied mathematics as an undergraduate at Christ Church, Oxford, the same college where Dodgson studied and had also been a Fellow.

Turing is known to have read Lewis Carroll’s 1886 book *The Game of Logic* [46] at school. Eperson wrote a book in 1948 on Lewis Carroll puzzles [72] and

Turing was writing for two copies of a new edition, enclosing a cheque for 7 shillings (35 UK pence). In the letter, Turing states that he has joined Manchester University and is working on “the use of electronic computers”. He mentions that there is the possibility of exhibiting a computer at the Festival of Britain (held in London during 1951). He opines [23]:

It is most entertaining work: one can make these machines do almost anything one wants, at any rate anything which one could explain rules for working out.

He goes on to mention a recent visit with his PhD student Robin Gandy to Cerne in Dorset, not far from Eperson, regretting that he did not know that Eperson was in Dorset at the time. Gandy’s parents lived in Dorset, a likely reason for the visit. Overall, the letter gives the impression that Turing is enjoying his new life based at Manchester University.

In summary, Turing’s memory has entered the cultural consciousness of the British public and the wider public around the world through the many faceted aspects of his life and work, bringing together a number of communities in a united respect for him. Artefacts such as manuscripts associated with Turing are increasingly valuable as a result, but are very limited in number. Although interest may have peaked especially for his 2012 centenary, it is likely to continue to increase over time.

7 Political impact

The British wartime leader Winston Churchill (1874–1965) recognized the role of Alan Turing and others at Bletchley Park in winning World War II as the “the geese who laid the golden eggs and never cackled” [113]. However, the secrecy around Bletchley Park meant that Turing only received an OBE award, whereas in other circumstances a knighthood could have been expected. Turing’s subsequent arrest for “gross indecency” in 1952 only lowered his esteem in political eyes after the War. It was not until much later after the decriminalisation of homosexuality in 1967 and the lifting of secrecy around Bletchley Park in the 1970s that it became possible for Turing to be truly politically acceptable and appreciated.

By 1998, at the unveiling of the blue plaque marking the birthplace of Turing in Maida Vale, west London (see Figure 6), during an oration by Turing’s biographer Andrew Hodges, Chris Smith, one of the first openly gay UK Members of Parliament, was able to send the following message [97]:

Alan Turing did more for his country and for the future of science than almost anyone. He was dishonourably persecuted during his life; today let us wipe that national shame clean by honouring him properly.

In a wider context, the 2006 book rebutting religion, *The God Delusion*, by the Oxford biologist and declared atheist Richard Dawkins includes mention



Fig. 6. The mathematician and Turing biographer, Andrew Hodges, unveiling the English Heritage blue plaque at Turing's birthplace, now the Colonnade Hotel, London, on 23 June 1998, exactly 86 years after Turing's birth. (Photograph by Jonathan Bowen.)

of Turing in the context of his homosexuality and persecution due to religious legacy [64]. Dawkins notes:

As the pivotal intellect in the breaking of the German Enigma codes, Turing arguably made a greater contribution to defeating the Nazis than Eisenhower or Churchill. . . . he should have been knighted and feted as a saviour of his nation. Instead this gentle, stammering, eccentric genius was destroyed, for a 'crime', committed in private, which harmed nobody.

The UK computer scientist John Graham-Cumming led a campaign for a governmental apology for Turing's treatment at the hands of British justice. The campaign received the backing of a wide range of people, including the scientist Richard Dawkins, the writer Ian McEwan, and the gay rights campaigner Peter Tatchell. In 2009, after a petition on the prime ministerial website, Gordon Brown, then Prime Minister of the UK, delivered an official apology from the British Government for the treatment of Alan Turing in his final years [9]. Brown wrote in *The Daily Telegraph* UK newspaper [41]:

While Turing was dealt with under the law of the time and we can't put the clock back, his treatment was of course utterly unfair and I am pleased to have the chance to say how deeply sorry I and we all are for

what happened to him. . . . This recognition of Alan’s status as one of Britain’s most famous victims of homophobia is another step towards equality, and long overdue.

In 2011, an e-petition on a UK government website requested an official pardon. The request was rejected in 2012, Turing’s centenary year, by Lord McNally, the Justice Secretary, who stated that Turing was “properly convicted” [10]. However, on 24 December 2013, after much campaigning, Queen Elizabeth II issued an official posthumous royal pardon for Alan Turing, in recognition for his wartime contributions [150]:

NOW KNOW YE that we, in consideration of circumstances humbly represented to us, are graciously pleased to grant our grace and mercy unto the said Alan Mathison Turing and grant him our free pardon posthumously in respect of the said convictions; AND to pardon and remit unto him the sentence imposed upon him as aforesaid; AND for so doing this shall be a sufficient Warrant.

It was granted under the Royal Prerogative of Mercy, on request from the UK Justice Minister at the time, Chris Grayling, who stated [11]:

Turing deserves to be remembered and recognised for his fantastic contribution to the war effort and his legacy to science. A pardon from the Queen is a fitting tribute to an exceptional man.

A general pardon for convicted homosexuals in the UK was delayed, no doubt due to the fear of compensation costs. However, in 2016, new law in England and Wales, through amendment of the *Policing and Crime Act*, was initiated, widely called the ‘Alan Turing law’ [13]. This received Royal Assent for approval as law on 31 January 2017 [14]. The new Justice Minister stated [120]:

This is a truly momentous day. We can never undo the hurt caused, but we have apologised and taken action to right these wrongs. I am immensely proud that ‘Turing’s Law’ has become a reality under this government.

Similar changes in Scottish law were announced later in 2017 [6]. It is to be hoped for the future that this could have a wider impact to laws in other countries around the world.

8 Conclusion

Andrew Hodges’ 1983 biography of Turing was updated with a new edition for the 2012 centenary [96] and has inspired a number of works, including a play, film, and music [30]. A more recent edited book in 2017, *The Turing Guide*, provides an accessible and comprehensive guide to Turing’s work [58]. This book has been of interest for a number of fields including computer science [107], philosophy

[132], physics [48], and even digital humanities [82]. Whereas the Hodges book is a single-author biography that covers Turing's work, [58] is an edited volume with 33 contributing authors covering the many aspects of Turing's work and research interests, complementing the biographical work [139].

Sadly, Turing lived too early for the change in the UK law to decriminalize homosexuality in 1967, but as Sir Dermot Turing, Alan Turing's nephew, notes in the conclusion of his own 2015 book on Turing [161]:

Alan Turing's life was not, except perhaps towards the end, governed by his sexuality. The dominant passion in his life was his ideas; it is those for which he should be remembered.

So, on a brighter note, Turing was selected as one of 43 top scientists of all time in the 2012 book *The Scientists* edited by Andrew Robinson [137], fortuitously published during the year of Turing's centenary. John von Neumann was also selected, but with an entry of half the length of that of Turing. Naturally the celebrated theoretical physicist Albert Einstein, who died in 1955, a year after Turing but following a much longer life, was also included in the book. It is interesting to speculate on the relative stature of Einstein and Turing, who overlapped in Princeton during the late 1930s and may have even met there, for example through Max Newman. Of course, Einstein has had a head start and is probably the most well-known scientist in the world, but as this paper has illustrated, Turing has been rapidly rising in the public's consciousness not only in the UK but worldwide. Alan Turing's place in the historical pantheon of scientists is now assured.

In the field of formal methods, although there is apparently no direct link from Turing's early interest in program proving, with hindsight it can be seen that Turing's ideas are highly relevant to this domain. Had Turing lived longer as the interests of other researchers caught up with those of Turing in the 1960s, it is quite possible that he could have been involved in the early formation of the formal methods community [31,38].

In 2019, the BBC held a public vote for the greatest person of the 20th century under a BBC Two programme series named *ICONS* [16]. This included seven categories presented week by week, namely leaders, explorers, scientists, entertainers, activists, sports, and artists/writers. As well as Turing, the short-listed scientists for selection were the Polish physicist and chemist Marie Curie (1867–1934), Albert Einstein, and the Chinese pharmaceutical chemist and 2015 Nobel Laureate Tu Youyou (屠呦呦, born 1930), known for a breakthrough in the treatment of malaria. Turing won the scientist category and went on to the final, against the finalists in the other categories, namely Muhammad Ali, David Bowie, Martin Luther King Jr, Nelson Mandela, Pablo Picasso, and Ernest Shackleton. The naturalist and broadcaster Chris Packham gave an impassioned and rousing speech supporting Alan Turing in the live final on 6 February 2019, which Turing duly won. Packham has Asperger's syndrome, so is able to empathise with Turing on both a scientific and personal level. He said that Turing was “a genius, a savour, but he was also autistic and gay” [17]. He continued:



Fig. 7. Posters displayed during the SETSS 2018 Spring School at Southwest University, Chongqing, including a young Turing, publicizing the ACM TURC 2018 Turing Celebration Conference in Shanghai, China, 19–20 May 2018 [3]. (Photograph by Jonathan Bowen.)

The scientists are the only ones that are going to save us and they are armed with Alan Turing’s legacy. [applause] I’ve got an idea. Get your phones out of your pocket, and turn them on, and hold them up, just so that they can sparkle. In each of your hands, you hold a little bit of Alan Turing. He’s with us when we wake up, he’s with us when we go to bed at night, and he’s with us when we talk to our loved ones. He’s beautiful, isn’t he, glistening in our darkest hour. You see, Alan Turing’s legacy hasn’t passed, he’s not a relic of the 20th century, his gift to us is our future. Thank you.

9 Postscript

In 1994, a formal methods meeting, the Z User Meeting, was held in Cambridge [34], 45 years to the week after the 1949 Cambridge meeting organized by Maurice Wilkes at which Turing presented his pioneering program proving paper [156]. The 1949 meeting was mentioned in the opening remarks of the 1994 Z User Meeting (ZUM) proceedings [26] and Wilkes, who led the Computer Laboratory in Cambridge until 1980, gave the after-dinner speech for the ZUM’94 conference. It could have been even more appropriate if Alan Turing, who would have been 82 years old in 1994 had he lived (almost exactly a year older than Wilkes), could have delivered the speech. Sadly of course he died aged 41, half the age that he would have been in 1994. We can only imagine, if he had lived

a further 41 years, how he could have contributed to the development of formal methods in particular and computer science in general – and most likely beyond, as we have seen in this paper that he did in any case, scientifically, culturally, and politically.

Acknowledgments: Parts of this paper are based on an earlier version of a book chapter [33]. Thank you especially to Troy Astarte, and also to Andrew Robinson and Tula Giannini, for comments on earlier drafts. The author is grateful to Southwest University and Museophile Limited for financial support in attending the SETSS 2018 Spring School (see Figure 7).

References

1. Abrial, J.-R.: The B-Book: Assigning Programs to Meanings. Cambridge University Press (1996)
2. ACM: ACM A.M. Turing centenary celebration. Association for Computing Machinery (2012). <http://turing100.acm.org>
3. ACM: ACM Turing Celebration Conference – China (ACM TURC 2018, ACM 中国图灵大会). Association for Computing Machinery (2018). <http://china.acm.org/TURC/2018/>
4. ACM: A.M. Turing Award. Association for Computing Machinery. <http://amturing.acm.org>
5. ACM: Chronological listing of A.M. Turing Award winners by year. Association for Computing Machinery. <http://amturing.acm.org/byyear.cfm>
6. Adams, L: Gay men to receive ‘Turing Law’ pardons. BBC News, BBC Scotland (31 August 2017). <http://www.bbc.com/news/uk-scotland-41108768>
7. Alton, J., Weiskittel H., Latham-Jackson, J.: Catalogue of the papers and correspondence of Christopher Strachey, 1930–1983. Department of Special Collections, Bodleian Library, University of Oxford, UK (2016). <http://www.bodley.ox.ac.uk/dept/scwmss/wmss/online/modern/strachey-c/strachey-c.html>
8. Augarten, S.: Bit by Bit: An Illustrated History of Computers, pp. 142–148. Ticknor & Fields (1984)
9. BBC: PM apology after Turing petition, BBC News (11 September 2009). <http://news.bbc.co.uk/1/hi/technology/8249792.stm>
10. BBC: Government rejects pardon request for Alan Turing. BBC News (8 March 2012). <http://www.bbc.co.uk/news/technology-16919012>
11. BBC: Royal pardon for codebreaker Alan Turing. BBC News (24 December 2013). <http://www.bbc.co.uk/news/technology-25495315>
12. BBC: Computer AI passes Turing test in ‘world first’. BBC News (9 June 2014). <http://www.bbc.co.uk/news/technology-27762088>
13. BBC: ‘Alan Turing law’: Thousands of gay men to be pardoned. BBC News (20 October 2016). <http://www.bbc.co.uk/news/uk-37711518>
14. BBC: Thousands of gay men pardoned for past convictions. BBC News (31 January 2017). <http://www.bbc.co.uk/news/uk-38814338>
15. BBC: Turing letters found in old filing cabinet. BBC News (19 August 2017). <http://www.bbc.co.uk/news/technology-41082391>
16. BBC: ICONS. BBC Two (2019). <http://www.bbc.co.uk/programmes/b0by86tp>
17. BBC: Chris Packham on Alan Turing. ICONS, BBC Two (6 February 2019). <http://https://www.bbc.co.uk/programmes/p0704h04>

18. Berma, P. Doolen, G.D. Mainieri, R. Tsifrinovich, V.I.: Turing machines. In: Introduction to Quantum Computers, chapter 2, pp. 8–12. World Scientific (1998)
19. Beth, E.W.: Semantic Entailment and Formal Derivability. Noord-Hollandsche (1955)
20. Beth, E.W.: Formal Methods. New York: Gordon & Breach (1962)
21. Blyth, T. (ed.): Information Age: Six Networks that Changed the World. Scala Arts & Heritage Publishers (2014)
22. Boca, P.P., Bowen, J.P., Siddiqi, J.I.: Formal Methods: State of the Art and New Directions. Springer (2010)
23. Bonhams: Lot 103 – Turing (Alan). Fine Books, Atlases, Manuscripts and Photographs. Bonhams, London (15 November 2017)
24. Bostrom, N.: Superintelligence: Paths, Dangers, Strategies. Oxford University Press (2014)
25. Boughey, C.L.F., et al.: Upper School, Form V a (Group III), Name Turing, Age 16. School report. Sherborne School, UK (Michaelmas Term 1928)
26. Bowen, J.P.: Introductory and opening remarks. In: Bowen, J.P., Hall, J.A. (eds.) [34], pp. v–vii (1994)
27. Bowen, J.P.: Computer science. In: Heilbron, J.L. (ed.) [85], pp. 171–174 (2003)
28. Bowen, J.P.: The Z notation: Whence the cause and whither the course? In: Liu, Z., Zhang, Z. (eds.) Engineering Trustworthy Software Systems, First International School, SETSS 2014. Springer, LNCS, vol. 9506, pp. 103–151 (2015). https://doi.org/10.1007/978-3-319-29628-9_3
29. Bowen, J.P.: Alan Turing: Virtuosity and visualisation. In: Bowen, J.P., Diprose, G., Lambert, N. (eds.) EVA London 2016: Electronic Visualisation and the Arts, pp. 197–205. BCS, Electronic Workshops in Computing (eWiC) (2016). <https://doi.org/10.14236/EVA2016.40>
30. Bowen, J.P.: Alan Turing: Founder of computer science. In: Bowen, J.P., Liu, Z., Zhang, Z. (eds.) Engineering Trustworthy Software Systems, Second International School, SETSS 2016. Springer, LNCS, vol. 10215, pp. 1–15 (2017). https://doi.org/10.1007/978-3-319-56841-6_1
31. Bowen, J.P.: Provably Correct Systems: Community, connections, and citations. In: Hinchey, M.G., et al. (eds.) [91], pp. 313–328 (2017)
32. Bowen, J.P., Angus, J., Bennett, J., Borda, A., Beler, A., Hodges, A., Filippini-Fantoni, S.: The development of science museum websites: Case studies. In: Hin, L.T.W., Subramaniam, R. (eds.) E-learning and Virtual Science Centers, chapter XVIII, pp. 366–392. Idea Group Publishing (2005)
33. Bowen, J.P., Copeland, B.J.: Turing’s legacy. In: Copeland, J., et al. [58], chapter 42, pp. 463–474 (2017)
34. Bowen, J.P., Hall, J.A. (eds.): Z User Workshop, Cambridge 1994. Springer, Workshops in Computing (1994)
35. Bowen, J.P., Hinchey, M.G., Glass, R.L.: Formal methods: Point-counterpoint. *Computer* **29**(4), 18–19 (April 1996)
36. Bowen, J.P., Hinchey, M.G.: Formal methods. In: Gonzalez, T., et al. (eds.) [152], part VIII, Programming Languages, chapter 71, pp. 71-1–71-25 (2014)
37. Bowen, J.P., Hinchey, M.G., Janicke, H., Ward, M., Zedan, H.: Formality, agility, security, and evolution in software engineering. In: Software Technology: 10 Years of Innovation in IEEE Computer, chapter 16, pp. 282–292. Wiley/IEEE Computer Society Press (2018)
38. Bowen, J.P., Reeves, S.: From a Community of Practice to a Body of Knowledge: A case study of the formal methods community. In: Butler, M., Schulte, W. (eds.)

- FM 2011: 17th International Symposium on Formal Methods. Springer, LNCS, vol. 6664, pp. 308–322 (2011). https://doi.org/10.1007/978-3-642-21437-0_24
39. Bowen, J.P., Trickett, T., Green, J.B.A., Lomas, A.: Turing’s genius – defining an apt microcosm. In: Bowen, J.P., Weinel, J., Diprose, G., Lambert, N. (eds.) *EVA London 2018: Electronic Visualisation and the Arts*, pp. 155–162. BCS, Electronic Workshops in Computing (eWiC) (2018). <https://doi.org/10.14236/EVA2018.31>
 40. Brattka, V.: Computability and Analysis, a Historical Approach. In: Beckmann, A., Bienvenu, L., Jonoska, N. (eds.) *Pursuit of the Universal*, CiE 2016. Springer, LNCS, vol. 9709 (2016). https://doi.org/10.1007/978-3-319-40189-8_5
 41. Brown, G.: I’m proud to say sorry to a real war hero. *The Telegraph* (10 September 2009).
 42. Bullynck, M., Daylight, E.G., De Mol, L.: Why did computer science make a hero out of Turing? *Communications of the ACM* **58**(3), 37–39 (March 2015). <https://doi.org/10.1145/2658985>
 43. Campagna, R., Cuomo, S., Giannino, F., Severino, G., Toraldo, G.: A semi-automatic numerical algorithm for Turing patterns formation in a reaction-diffusion model. *IEEE Access* **6**, 4720–4724 (6 December 2017). <https://doi.org/10.1109/ACCESS.2017.2780324>
 44. Campbell, S.: Beatrice Helen Worsley: Canada’s female computer pioneer. *IEEE Annals of the History of Computing* **25**(4), 51–62 (October–December 2003). <https://doi.org/10.1109/MAHC.2003.1253890>
 45. Campbell-Kelly, M., Aspray, W., Ensmenger, N., Yost, J.R.: *Computer: A History of the Information Machine*, 3rd edition. Westview Press (2014)
 46. Carroll, L.: *The Game of Logic*. London: Macmillan & Co. (1886). <http://archive.org/details/gameoflogic00carrich>
 47. Carus, P.: Formal thought the basis of kenlore. *The Monist* **20**(4), 574–584 (October 1910). <https://doi.org/10.5840/monist191020428>
 48. Cerf, V.: The man behind the machine. *Physics World*, pp. 38–39 (January 2018). <http://physicsworld.com/a/the-man-behind-the-machine/>
 49. Collins, R.: Steve Jobs review: ‘manically entertaining’. *The Telegraph* (12 November 2015). <http://www.telegraph.co.uk/film/steve-jobs/review>
 50. Cooper, S.B.: Turing’s Titanic Machine? *Communications of the ACM* **55**(3), 74–83 (March 2012). <https://doi.org/10.1145/2093548.2093569>
 51. Cooper, S.B.: *The Alan Turing Year: A Centenary Celebration of the Life and Work of Alan Turing*. School of Mathematics, University of Leeds, UK (2012). <http://www.turingcentenary.eu>
 52. Cooper, S.B., van Leeuwen, J. (eds.): *Alan Turing: His Work and Impact*. Elsevier Science (2013)
 53. Copeland, B.J. (ed.): *Colossus: The Secrets of Bletchley Park’s Codebreaking Computers*. Oxford University Press (2006)
 54. Copeland, B.J. (ed.): *The Essential Turing*. Oxford University Press (2004)
 55. Copeland, B.J.: Artificial intelligence. In: Copeland, B.J. (ed.) [54], pp. 353–361 (2004)
 56. Copeland, B.J.: *Turing: Pioneer of the Information Age*. Oxford University Press (2012)
 57. Copeland, B.J.: Alan Turing: The codebreaker who saved ‘millions of lives’. *BBC News* (19 June 2012). <http://www.bbc.co.uk/news/technology-18419691>
 58. Copeland, B.J., Bowen, J.P., Sprevak, M., Wilson, R., et al.: *The Turing Guide*. Oxford University Press (2017)
 59. Copeland, B.J., Posy, C.J., Shagrir, O.: *Computability: Turing, Gödel, Church, and Beyond*. MIT Press (2013)

60. Copeland, B.J., Proudfoot, D.: The Turing Archive for the History of Computing. <http://www.alanturing.net>
61. Dahl, O.-J., Dijkstra, E.W., Hoare, C.A.R.: Structured Programming. Academic Press (1972)
62. Dasgupta, S.: It Began with Babbage: The Genesis of Computer Science. Oxford University Press (2014)
63. Davis, M.: Turing's Vision and deep learning. In: Manea, F., Miller, R., Nowotka, D. (eds.) Sailing Routes in the World of Computation, 14th Conference on Computability in Europe, CiE 2018. Springer, LNCS, vol. 10936, pp. 146–155 (2018). https://doi.org/10.1007/978-3-319-94418-0_15
64. Dawkins, R.: The God Delusion, p. 289. Bantam Press (2006)
65. Daylight, E.G.: Towards a Historical Notion of 'Turing—the Father of Computer Science'. *History and Philosophy of Logic* **36**(3), pp. 205–228 (2015). <https://doi.org/10.1080/01445340.2015.1082050>
66. De Mol, L.: Turing Machines. In: Zalta, E.N. (ed.) *Stanford Encyclopedia of Philosophy*. Stanford University, USA (24 September 2018). <http://plato.stanford.edu/entries/turing-machine>
67. Deutsch, D.: Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A* **400**, 97–117 (1985). <https://doi.org/10.1098/rspa.1985.0070>
68. Dijkstra, E.W.: A Discipline of Programming. Prentice Hall (1976)
69. Dunham, W.: Euler: The Master of Us All. Mathematical Association of America (1999)
70. Dyson, G.: Darwin Among the Machines. The Penguin Press (1997)
71. Edwards, J.R.: An early history of computing at Princeton. Princeton Alumni Weekly (4 April 2012). <http://paw.princeton.edu/article/early-history-computing-princeton>
72. Eperson, D.B.: The Lewis Carroll Puzzle Book: Containing over 1,000 posers from Alice in Wonderland and other books by Lewis Carroll. Appeal Office (1948)
73. Evans, C.: The Mighty Micro: The Impact of the Computer Revolution. Victor Gollancz (1979)
74. Feigenbaum, E.A., McCorduck, P.: The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World. Addison Wesley (1983)
75. Floyd, R.W.: Assigning meaning to programs. In: Schwartz, S.T. (ed.) *Mathematical Aspects of Computer Science*. American Mathematical Society (1967). https://doi.org/10.1007/978-94-011-1793-7_4
76. Frith, H.: Unraveling the tale behind the Apple logo. CNN (7 October 2011). <http://edition.cnn.com/2011/10/06/opinion/apple-logo>
77. Giannini, T., Bowen, J.P.: Life in code and digits: When Shannon met Turing. In: Bowen, J.P., Diprose, G., Lambert, N. (eds.) *EVA London 2017: Electronic Visualisation and the Arts*, pp. 51–58. BCS, Electronic Workshops in Computing (eWiC) (2017). <https://doi.org/10.14236/EVA2017.9>
78. Google: Alan Turing. Google Scholar. <http://scholar.google.com/citations?user=VWCHLwkAAAAJ>
79. Grattan-Guinness, I.: Turing's mentor, Max Newman. In: Copeland, B.J. et al. [58], chapter 40, pp. 437–442 (2017)
80. Gray, P.: Computer scientist: Alan Turing. *Time* **153**(12) (29 March 1999)
81. Gries, D.: The Science of Programming. Springer, Texts and Monographs in Computer Science. (1981)

82. Han, B.: 枷锁与馈赠|认识真实的图灵, 一位如谜的解谜者 [trans.: Shackles and gifts – Know the real Turing, a mystery puzzler]. WeChat (3 April 2018). <http://mp.weixin.qq.com/s/JcdromoslivadmFmEf8SVQ>
83. Harel, D.: *Computers Ltd.: What they really can't do*. Oxford University Press (2000)
84. Hayes, P.J., Morgenstern, L.: On John McCarthy's 80th birthday, in honor of his contributions, *AI Magazine* **28**(4), 93–102 (Winter 2007). <https://doi.org/10.1609/aimag.v28i4.2063>
85. Heilbron, J.L. (ed.): *The Oxford Companion to the History of Modern Science*. Oxford University Press (2003)
86. Hickey, S.: Alan Turing notebook sells for more than \$1m at New York auction. *The Guardian* (13 April 2015)
87. Highfield, R.: Codebreaker wins Great Exhibition Award. Science Museum, London (17 December 2012). <http://blog.sciencemuseum.org.uk/codebreaker-wins-great-exhibition-award>
88. Highfield, R.: What to think about machines that think. Science Museum, London (11 December 2015). <http://blog.sciencemuseum.org.uk/what-to-think-about-machines-that-think>
89. Hillis, D., McCarthy, J., Mitchell, T.M., Mueller, E.T., Riecken, D., Sloman, A., Winston, P.H.: In honor of Marvin Minsky's contributions on his 80th birthday, *AI Magazine* **28**(4), 103–110 (Winter 2007). <https://doi.org/10.1609/aimag.v28i4.2064>
90. Hilton, P.: Meeting a genius. In: Copeland, J., et al. [58], chapter 3, pp. 31–34 (2017)
91. Hinchey, M.G., Bowen, J.P., Olderog, E.-R. (eds.): *Provably Correct Systems*. Springer, NASA Monographs in Systems and Software Engineering (2017)
92. Hinchey, M.G., Coyle, L. (eds.): *Conquering Complexity*. Springer (2012)
93. Hoare, C.A.R.: An axiomatic basis for computer programming, *Communications of the ACM*, **12**(10), 576–580 (October 1969). <https://doi.org/10.1145/363235.363259>
94. Hoare, C.A.R., Hayes, I.J., He, J., Morgan, C.C., Roscoe, A.W., Sanders, J.W., Sørensen, I.H., Spivey, J.M., Sufrin, B.: Laws of programming. *Communications of the ACM* **30**(8), pp. 672–686 (August 1987). <https://doi.org/10.1145/27651.27653>
95. Hoare, C.A.R., He, J.: *Unifying Theories of Programming*. Prentice Hall International Series in Computer Science (1998)
96. Hodges, A.: *Alan Turing: The Enigma*. Burnett / Simon and Schuster (1983); Centenary edition, Princeton University Press (2012)
97. Hodges, A.: Oration at Alan Turing's birthplace. *Alan Turing: The Enigma* (23 June 1998). <http://www.turing.org.uk/publications/oration.html>
98. Hodges, A.: *Alan Turing: The Enigma*. <http://www.turing.org.uk>
99. IMDb: *The Imitation Game*. IMDb (2014). <http://www.imdb.com/title/tt2084970>
100. Isaacson, W.: *Steve Jobs*, p. xvi. Simon & Schuster / Little, Brown (2011)
101. Jones, A.: Five 1951 BBC broadcasts on automatic calculating machines. *IEEE Annals of the History of Computing* **26**(2), pp. 3–15 (April–June 2004). <https://doi.org/10.1109/MAHC.2004.1299654>
102. Jones, A.: Brains, tortoises, and octopuses: Postwar interpretations of mechanical intelligence on the BBC. *Information & Culture* **51**(1), pp. 81–101 (2016). <https://doi.org/10.7560/IC51104>
103. Jones, C.B.: *Systematic Software Development Using VDM*. Prentice Hall International Series in Computer Science (1986)

104. Jones, C.B.: Turing's "Checking a Large Routine". In: Cooper, S.B., van Leeuwen, J. (eds.) *Alan Turing – His Work and Impact*, pp. 455–461. Elsevier (2013)
105. Jones, C.B.: Turing and Software Verification. Technical Report CS-TR-1441, Newcastle University (December 2014). <http://homepages.cs.ncl.ac.uk/cliff.jones/publications/NU-TRs/CS-TR-1441.pdf>
106. Jones, C.B.: Turing's 1949 paper in context. In: Kari, J., Manea, F., Petre, I. (eds.) *Computability in Europe 2017*. Springer, LNCS, vol. 10307, pp. 21–41 (2017). https://doi.org/10.1007/978-3-319-58741-7_4
107. Jones, C.B.: The Turing Guide. *Formal Aspects of Computing* **29**, 1121–1122 (2017). <https://doi.org/10.1007/s00165-017-0446-y>
108. Kettle, S.: Alan Turing. <http://www.stephenkettle.co.uk/turing.html>
109. King's College: The Turing Digital Archive. King's College, Cambridge, UK. <http://www.turingarchive.org>
110. Lavington, S.: The ACE, the 'British National Computer'. *Early British Computers*, chapter 5, pp. 23–30. Manchester University Press (1980)
111. Lavington, S. (ed.): *Alan Turing and his Contemporaries: Building the World's First Computers*. BCS, The Chartered Institute for IT (2012)
112. Lavington, S.: Hindsight and foresight: The legacy of Turing and his contemporaries. In: Lavington, S., (ed.) [111], chapter 8, pp. 79–84 (2012)
113. Lewin, R.: *Ultra Goes to War*, p. 64. Grafton (1978)
114. Manchester University: Lost Turing letters give unique insight into his academic life prior to death. *Discover / News*. The University of Manchester, UK (25 August 2017). <http://www.manchester.ac.uk/discover/news/>
115. McCarthy, J.: A basis for a mathematical theory of computation. In: Braffort P., Hirschberg, D. (eds.) *Computer Programming and Formal Systems*, pp. 33–70. North-Holland (1963). [https://doi.org/10.1016/S0049-237X\(08\)72018-4](https://doi.org/10.1016/S0049-237X(08)72018-4)
116. McKay S.: *The Secret Life of Bletchley Park*. Aurum (2011)
117. McKay, S.: How Alan Turing's secret papers were saved for the nation. *The Telegraph* (30 July 2011)
118. MGP: Alan Mathison Turing. Mathematics Genealogy Project. Department of Mathematics, North Dakota State University, USA. <http://www.genealogy.ams.org/id.php?id=8014>
119. Millican, P.J.A., Clark, A. (eds.): *The Legacy of Alan Turing*. Oxford University Press. Volume I: *Machines and Thought*; Volume II: *Connectionism, Concepts and Folk Psychology* (1996)
120. Ministry of Justice, Gyimah, S.: Thousands officially pardoned under 'Turing's Law'. UK Government (31 January 2017). <http://www.gov.uk/government/news/thousands-officially-pardoned-under-turings-law>
121. Morgan, C.C.: *Programming from Specifications*. Prentice Hall International Series in Computer Science (1990). 2nd edition (1994)
122. Morris, F.L., Jones, C.B.: An early program proof by Alan Turing. *IEEE Annals of the History of Computing* **6**(2), 139–143 (April 1984). <https://doi.org/10.1109/MAHC.1984.10017>
123. Nature: Alan Turing at 100. *Nature* **482**, 450–465 (25 February 2012) <http://www.nature.com/news/specials/turing>
124. Naur, P.: Proof of algorithms by general snapshots. *BIT* **6**, 310–316 (1966). <https://doi.org/10.1007/BF01>
125. Newman, M.H.A.: Alan Mathison Turing, 1912–1954. *Biographical Memoirs of Fellows of the Royal Society* **1**, 253–263 (November 1955). <https://doi.org/10.1098/rsbm.1955.0019>

126. Newman, W.: Max Newman—Mathematician, codebreaker, and computer pioneer. In: Copeland, B.J. (ed.) [53], chapter 14, pp. 176–188 (2006)
127. O’Connor, J.J., Robertson, E.F.: Donald Birkby Eperson. MacTutor. School of Mathematics and Statistics, University of St Andrews, Scotland (May 2017). <http://www-history.mcs.st-and.ac.uk/Biographies/Eperson.html>
128. Page C., Richards, M.: A letter from Christopher Strachey. Resurrection: The Journal of the Computer Conservation Society **73**, 22–24 (Spring 2016). <http://www.computerconservationsociety.org/resurrection/res73.htm#d>
129. Palfreman, J., Swade, D.: The Dream Machine: Exploring the Computer Age. BBC Books (1991)
130. Penrose, R.: The Emperor’s New Mind: Concerning Computer, Minds, and the Laws of Physics. Oxford University Press (1989)
131. Penrose, R.: Shadows of the Mind: A Search for the Missing Science of Consciousness. Oxford University Press (1994)
132. Petrocelli, C.: The Turing Guide, by Jack Copeland, Jonathan Bowen, Mark Sprevak, and Robin Wilson. Nuncius **33**(1), 166–168. <https://doi.org/10.1163/18253911-03301015>
133. Princeton: Alan M.Turing. Office of the Executive Vice President, Princeton University, USA (2018). <http://evp.princeton.edu/people/alan-m-turing>
134. Randell, B.: On Alan Turing and the origins of digital computers. In: Meltzer, B., Michie, D. (eds.) Machine Intelligence, vol. 7, pp. 3–20. Edinburgh University Press (1972). <http://www.cs.ncl.ac.uk/research/pubs/books/papers/126.pdf>
135. Rendell, P.: Game of Life – Universal Turing Machine, YouTube (2010, uploaded 2012). <http://www.youtube.com/watch?v=My8AsV7bA94>
136. Robinson, A.: Sudden Genius: The Gradual Path to Creative Breakthroughs. Oxford University Press (2010)
137. Robinson, A. (ed.): The Scientists: An Epic of Discovery. Thames & Hudson (2012)
138. Robinson, A.: Film: Reality and check. The Lancet **386**, 2048 (21 November 2015)
139. Robinson, A.: The Turing Guide: Last words on an enigmatic codebreaker? New Scientist **3107**, 42–43 (4 January 2017). <http://www.newscientist.com/article/mg23331072-700>
140. Rooney, D.: Codebreaker – Alan Turing’s life and legacy. Science Museum, London. YouTube (19 June 2012). http://www.youtube.com/watch?v=I3NkVMHhO_Q
141. Rooney, D.: Mathematics: How it Shaped our World. Scala Arts & Heritage Publishers (2016)
142. Russell, B.: What is logic. In: The Collected Papers of Bertrand Russell. Vol. 6: Logical and Philosophical Papers, 1909–13. Part I: Logic and the Philosophy of Mathematics. Routledge (1912)
143. Shallis, M.: The Silicon Idol: The Micro Revolution and its Social Implications. Oxford University Press (1984)
144. Singh, S.: The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. London: Forth Estate (1999)
145. Smith, C.N., et al.: Form IV b (i), Name Turing, Average Age 14.6, Age 14.8. School report. Sherborne School, UK (Lent Term 1927)
146. Soare, R.I.: History of Computability. In: Turing Computability: Theory and Applications. Theory and Applications of Computability series, Springer (2016). https://doi.org/10.1007/978-3-642-31933-4_17
147. Srinivasan, A. (ed.): Donald Michie: On Machine Intelligence, Biology & More. Oxford University Press (2009)

148. Stoy, J.E.: Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory. MIT Press (1977)
149. Sumner, J.: Turing today. Notes and Records of the Royal Society of London **66**(3), pp. 295–300 (20 September 2012). <http://www.jstor.org/stable/41723310>
150. Swinford, S.: Alan Turing granted Royal pardon by the Queen. The Telegraph (24 September 2013)
151. Time: The great minds of the century. Time **153**(12) (29 March 1999). <http://content.time.com/time/magazine/article/0,9171,990608,00.html>
152. Gonzalez, T., Diaz-Herrera, J., Tucker, A.B. (eds.): Computing Handbook, 3rd edition. Volume I: Computer Science and Software Engineering. Chapman and Hall / CRC Press (2014)
153. Turing, A.M.: Précis of The Theory of Relativity by Albert Einstein. In: The Turing Digital Archive [109], AMT/K/2 (1927). <http://www.turingarchive.org/viewer/?id=449&title=1>
154. Turing, A.M.: On computable numbers with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society (Series 2) **42**(1), 230–265 (1936). <https://doi.org/10.1112/plms/s2-42.1.230>
155. Turing, A.M.: The Purpose of Ordinal Logics. PhD thesis, Princeton University, USA (1938)
156. Turing, A.M.: Checking a large routine. In: Report of a Conference on High Speed Automatic Calculating Machines, pp. 67–69. Mathematical Laboratory, University of Cambridge, UK (1949). <http://www.turingarchive.org/browse.php/b/8>
157. Turing, A.M.: Computing machinery and intelligence. Mind **59**(236), 433–460 (October 1950). <https://doi.org/10.1093/mind/LIX.236.433>
158. Turing, A.M.: Can Digital Computers Think? In: The Turing Digital Archive [109], AMT/B/5 (May 1951). <http://www.turingarchive.org/viewer/?id=449&title=1> (see also 2018 transcription under <http://aperiodical.com/wp-content/uploads/2018/01/Turing-Can-Computers-Think.pdf>)
159. Turing, A.M.: The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society of London **237**(641), 37–72 (1952). <https://doi.org/10.1098/rstb.1952.0012>
160. Turing, A.M., Braithwaite, R.B., Jefferson, G., Newman, M.: Can automatic calculating machines be said to think? (1952). In: Copeland, B.J. (ed.) [54], chapter 14, pp. 487–506 (2004).
161. Turing, D.: Prof Alan Turing Decoded: A Biography. The History Press (2015)
162. Turing, S.: Alan M. Turing: Centenary Edition. Cambridge University Press (2012)
163. Uhlig, R.: Genius of Britain: The Scientists who Changed the World. Harper-Collins (2010)
164. van Doren, C.: A History of Knowledge: Past, Present, and Future. Ballantine Books (1991)
165. Vardi, M.Y.: Would Turing have won the Turing Award? Communications of the ACM, **60**(11), p. 7 (November 2017). <https://doi.org/10.1145/3144590>
166. Veblen, O.: Letter to M. H. A. Newman, esq. Janus Catalogue Item 2-1-13. The Max Newman Digital Archive, University of Brighton, UK (4 May 1928). <http://www.cdpa.co.uk/Newman/MHAN>
167. Wikipedia: List of things named after Alan Turing. Wikipedia, Wikimedia Foundation. http://en.wikipedia.org/wiki/List_of_things_named_after_Alan_Turing

168. Wilkes, M.V.: Automatic calculating machines. *Journal of the Royal Society of Arts* **100**(4862), pp. 56–90 (14 December 1951). <http://www.jstor.org/stable/41365298>
169. Wolfram, S.: *A New Kind of Science*. Wolfram Media (2002)
170. Worsley, B.H.: *Serial Programming for Real and Idealised Digital Calculating Machines*. PhD thesis, University of Cambridge, UK (May 1952). Also in: Archives Center, National Museum of American History, Smithsonian Institution, USA. <http://sova.si.edu/details/NMAH.AC.0237#ref29>
171. Yates, D.M.: *Turing's Legacy: A history of computing at the National Physical Laboratory 1945–1995*. London: Science Museum (1997)