Alan Turing: the logical and physical basis of computing

Andrew Hodges*
Wadham College, University of Oxford, Oxford OX1 3PN, U.K.

This paper is based on the talk given on 5 June 2004 at the conference at Manchester University marking the 50th anniversary of Alan Turing's death. It is published by the British Computer Society on http://www.bcs.org/ewics. *It was submitted in April 2006; some corrections have been made and references added for publication in November 2007.*

1 Computable numbers

HALF a century after Alan Turing's death in 1954, we can ask whether computing can go beyond the framework of computability that he set out in his classic 1936 paper *On computable numbers, with an application to the Entscheidungsproblem.* We shall do so in the light of what Turing himself did, starting with his work in 1936.

It is usual to describe the background to that paper (Turing 1936a) in terms of Hilbert's decision problem for first-order logic, Gödel's discovery, and so forth. This is a perfectly correct description of how Turing came to this work through Newman's 1935 Cambridge lectures. But Turing made a typically individualistic point of setting forth his new definition of computability in terms of *numbers,* presenting the Hilbert problem as 'an application' of this definition. In what follows, this relationship of computability to number structure will be emphasised, because it is highly relevant to current proposals concerning the potential scope of computing. Of course, it is most important that from that 1936 start, Turing was always considering calculation within general symbolic systems, without restriction to numerical data. His practical proposals for digital computing were streets ahead in their recognition of non-numerical structures. But for the connection with physical reality, which we shall emphasise, the basis of computability in the classical mathematics of the integers $\mathbf{Z}$ and the real continuum $\mathbf{R}$ is essential.

Indeed, a clear picture of $\mathbf{Z}$ and $\mathbf{R}$ is necessary even to understand the title of Turing's paper. Turing's word 'numbers' refers to elements of $\mathbf{R}$: the 'real numbers', generally thought of as corresponding to the measurement of continuous quantities like distance and time, and expressible as infinite decimal fractions.[1] But the computation of

*email andrew.hodges@wadh.ox.ac.uk, web www.turing.org.uk

[1]The word 'decimal' properly implies an expansion specifically in inverse powers of *ten,* but it makes no difference, in the material discussed here, what number base is used for representations. In writing of 'decimals' we follow Turing, who introduced his own 1936 definition as: 'a number is computable if its decimal can be written down by a machine.'

a real number, in Turing's picture, is specified by defining a finite 'machine', consisting of certain discrete elements, which can all be coded as an element of **Z**, the 'description number' of the machine. That (some) elements of **R** can be encoded systematically by elements of **Z** is a non-trivial insight.

Students may well be puzzled that although they are taught that a well-defined algorithm must terminate, and that a Turing machine formalizes the concept of an algorithm, the computation of a computable number by a Turing machine never terminates! But there is no contradiction here. Turing's model for computing a real number demands that the calculation of *any one digit* is indeed guaranteed to be a terminating task. (No preset bound can generally be placed on the time it will take, however, nor on the amount of working space required.) Computation of the computable number corresponds to the calculation of each digit successively, and this of course cannot be done in a finite time. The crucial point is that the same finite package of information, which can be coded as a single 'description number', suffices for infinitely many different digits.

What Turing called the 'circle-free' condition is the condition that every digit can thus be calculated, and his theory demonstrated the undecidability of circle-free-ness. It is more common now to use the simpler 'halting problem'—the question of whether a given Turing machine, starting on a blank tape, will ever reach a particular 'halting' state—as a classic example of an undecidable problem. But this reformulation is of no fundamental significance.

Since the description of a computable number is given as an integer, the computable numbers are countable.[2] In this light, we can construct a picture of the 'real line' **R** with certain special countable subsets of points: first the integers, then the larger set of rationals, then the still larger set of algebraic numbers. The countable set of the computable numbers appears to be the final natural extension of this idea. It characterizes by quite simple means all the numbers used in vast areas of mathematics (as Turing showed in his 1936 paper) and this alone is a great achievement of Turing's. He hoped for yet more: Turing's introduction stated that he hoped soon to base 'the theory of functions of a real variable' on this concept, and this remark suggests that in 1936 he had every intention of redefining the relationship between **Z** and **R** in a constructivist way.

The structure of **R** does not enjoy the same intuitive and compelling status as that of **Z**, and Turing might well have felt there was room for improvement. A viable logical relationship between counting (**Z**) and measuring (**R**) was only elucidated in the nineteenth century, with notable steps taken by Weierstrass in 1860 and Dedekind in 1872. When Turing started to read mathematics at Cambridge in 1931, it had only been established there for a generation. (According to Ray Monk (1996), Bertrand Russell had graduated in mathematics in 1893 while completely unaware of it, and only learnt the modern definition of a limit in 1896.) G. H. Hardy's *Pure Mathematics*, first published in 1908, was the most famous text through which English mathematical students learnt the analysis of **R** developed in nineteenth-century Europe, but Turing actually studied

[2]More accurately, for any particular computable number there are infinitely many Turing machines, and so infinitely many integers.

the deeper treatment in the textbook of Knopp (1921). Knopp also emphasised the actual numerical computation of series, which made it a good introduction to Turing's down-to-earth treatment of computability.

It is a pity that Turing left no indication of how he developed his ideas in 1935. The computable numbers seem to come fully-formed without precursor or rehearsal in his work. However, a curious fact about Turing's papers is worth mentioning. They include both a manuscript and a typescript of a paper on normal numbers (Turing 1936?b), a topic springing from measure theory. Normal numbers are, roughly speaking, those in which the digits of the decimal expansion are uniformly distributed, no matter what number base is used. It was known that almost all numbers are normal, yet no explicit example of one had been given. But Turing's friend David Champernowne, also a Scholar at King's College, Cambridge, made a contribution to the theory, and so had a mathematical paper published while he was a second-year undergraduate. Champernowne (1933) wrote down the number 0.12345678910111213... and proved that it is normal in base 10. Turing's investigation may be seen as an attempt to give an equally explicit construction for more general normal numbers. Turing's proof had several faults, according to Britton (1992), and anyway was never published.

What lends extra interest to this piece of research is that Turing's manuscript was written on the reverse side of six pages from the typescript of *On computable numbers*. (The typing of the text is not his own, but the equations are written in his own hand.) The six pages include the section on 'computable convergence', which is the material that Turing might have hoped to expand into a constructivist definition of real analysis. One of these pages has now been published in a facsimile form, with further commentary (Hodges 2006b). Interestingly also, Turing's analysis of normal numbers includes a reference to 'a mechanical process' and 'constructive enumeration', indicating a link with computability. There is no indication of the date of this work, except that it appears on the typescript of *On computable numbers*, which places it in 1936 or later. But Turing had known of Champernowne's work since 1933. It seems possible that friendly rivalry with Champernowne in 1933-1935 gave Turing a particular personal motivation to think about defining real numbers in a constructive way *before* learning from Newman of the Entscheidungsproblem. It might also have increased his confidence to plunge into a research field, despite being young and a complete outsider. But the deeper philosophical interest evidenced by Turing's essay *Nature of Spirit* (Turing 1932?) seems more important as a precursor. Above all, it is due to Turing's quite inexplicable genius that he saw a connection, invisible to others, between Hilbert's decision problem, the classical question of free will and physical determinism, and the computation of real numbers.[3]

[3]Britton (1992) only edits and comments on Turing's typescript, which omits the reference to Champernowne. Britton comments that Champernowne had given an example of a normal number, but Turing's manuscript points out that Champernowne's number is only shown to be normal in base 10. Incidentally, it is sometimes believed by computer historians that binary representation marks a major step. For engineers, perhaps it was. But for mathematicians the possibility of representations in any base was a matter of common currency. If explicit proof were needed, Turing's usage in this paper would supply it.

Since the real numbers are uncountable, almost all the real numbers are uncomputable. Turing gave an explicit example of a definable but uncomputable number: the number which differs in the nth decimal place from the nth computable number, as ordered by their description numbers. Another uncomputable number Ω, related to the 'halting problem', has been made popular by Chaitin.[4] It would, however, be rather misleading to present Ω as a natural constant like π, because the definition depends on conventional technical details of how Turing machines are specified. It is also somewhat misleading for a deeper reason: whereas π, being a computable number, can be specified by a finite set of data, a number like Ω cannot. Some writers make much of how the infinite data in Ω, which corresponds to an infinite list of answers to deep mathematical questions, is encoded in a 'single' real number. But this is less surprising when it is remembered that the concept of a 'single' real number involves an infinite quantity of discrete data.

Turing thus opened up new aspects of the number system. But the novel technical apparatus he developed for doing this was equally important, because it has turned into the dominant technology of the modern world. Gödel had shown how to encode *as* integers, statements *about* integers, and thereby showed the incompleteness of formal systems. Turing showed how to encode operations *on* integers, *by* integers, and thereby showed the undecidability of formal systems. Thus Turing had the vital perception that operations and numerical data could be codified alike as a symbolic structure. This perception was essential for his proofs, but it was not limited to the abstruse material of the Entscheidungsproblem. For, in a remarkable application of that perception, Turing showed that 'it is possible to invent a single machine which can be used to compute any computable sequence'. This invention was his *universal machine.*

2 Turing's practical development of computability

Next we will briefly survey how Turing developed his own 1936 discovery, with an emphasis on the way he turned the logical into the physical. The most striking fact is that he later turned the concept of the universal machine into a practical prospectus for the digital computer. Turing had a conversation with his friend David Champernowne in the 1936 period discussing the practicality (or impracticality) of the universal machine, in the course of which the question of a machine the size of 'the Albert Hall' was raised. It is reasonable to suppose that Turing saw right away that the very simple logical operations of his machines could in principle be realized physically, since they required only operations such as were employed in contemporary teleprinters and automatic telephone exchanges. But a great deal had to happen before the idea of the universal machine could be made into a practical ambition.

In the period from 1937 to 1939, Turing followed many paths in parallel and no observer at the time could have seen that they led towards the electronic computer. We may note: (1) He did not not recast real analysis in terms of computable numbers

[4]For a recent example see (Chaitin 2006)

as he hoped. The problem he addressed in (Turing 1937), viz. the non-uniqueness of decimal expansions ($.50000\ldots = .49999\ldots$), is at first sight trivial, but in fact points to a deep difficulty: that no characteristic function of a proper subset of $\mathbf{R}$ is computable. The different description of $\mathbf{R}$ used by Turing (1937) makes the first step in modern computable analysis, but Turing himself took this no further. (2) Turing's Princeton doctoral thesis work (Turing 1938a) was concerned with the much deeper theory of uncomputable structures (now called Turing degrees), and not at all with physical or practical developments. (3) His work on the Riemann zeta-function (Turing 1943) and related questions in number theory was marked by an interest in actual computation. But the machine he designed to assist calculation (Turing 1939) was an analogue device with discrete approximations, not a development of his logical machine ideas. (4) His work in group theory (Turing 1938b, 1938c) was probably what attracted the offer from von Neumann of a post-doctoral position, which he declined. There is no reason to suppose that he had any discussion of computation with von Neumann. (5) Although joining Wittgenstein's class on the philosophy of mathematics, Turing's remarks did not reflect his computational ideas, nor a philosophy of mind such as he later espoused. (6) As a non-academic sideline, he built electric relay machinery to perform binary multiplication. This was specifically intended for an advanced cipher system, which he may have derived from considering the question of 'the most general code or cipher'. This offbeat amateur engineering was the closest Turing came to developing his ideas of general computation in a practical direction.

As it turned out, it was also this eccentric hobby—as it would then have seemed—that came to dominate his life from 1939 to 1945, and thereby arguably changed world history. In 1948 Turing wrote of how machinery was generally considered as limited to straightforward, repetitive tasks until 'about 1940', and it can hardly be doubted that by citing that date he was alluding to the impact of the Turing-Welchman Bombe with its stunning parallel logic for solving the plugboard complication of the Enigma enciphering machine used for German military communications. From that time onwards, Turing led the way at Bletchley Park in showing the power of computable operations. Some of these were implemented on physical machines, others through people carrying out algorithmic processes such as the 'Banburismus' that exploited his new Bayesian inference methods (Good 1992, 1993, 2001). Turing's mechanization of judgment through 'weight of evidence' (anticipating statistical methods very popular in Artificial Intelligence sixty years later) was perhaps a key step in persuading him of the boundless scope for the mechanization of human intelligence. In 1943, he discussed such ideas with Shannon, and in 1944 he was speaking of 'building a brain'. The possibilities for actual engineering had been transformed since the 'Albert Hall' days of 1936, with the availability and proved practicality of digital electronic techniques.

The use of the word 'brain' indicated that at some point Turing had decided that the scope of *On computable numbers* could and should be broadened. Whilst in 1936 he had limited the discussion to that of a human being carrying out a mechanical process, from 1945 onwards he took the view that *all* mental processes fell within the scope of computable operations. The ground had already been laid in 1936, by the bold argument

that a process need not actually be written down in 'instruction notes', but could be embodied in 'states of mind'. But Turing's post-1945 views still marked a definite shift. In particular they implied a rejection of the view that uncomputable 'intuitive' steps were needed for the recognition of true but formally unprovable Gödel statements. Such uncomputable mental steps were implied by his 1938 analysis, and Gödel himself certainly believed in them. One sentence in Turing's proposal for the ACE computer (Turing 1946) is evidence of this shift: it is a claim that a computer could play 'very good chess' if 'occasional serious mistakes' were allowed. This cryptic reference to mistakes was clarified by the argument of (Turing 1947) that once the possibility of mistakes is allowed, Gödel's theorem becomes irrelevant. In Turing's post-war view, 'seeing the truth' is done, imperfectly, by algorithms; and there is no reason why computers should not do it as successfully as human beings. (Gödel, of course, rejected this post-war Turing vision of machine intelligence, although he had endorsed Turing's original definition of computability.) Turing's use of the word 'brain' also indicated an interest in its *physical* nature, which we will discuss further below.

This deep interest in the nature of mental operations was Turing's principal motivation, but in 1945 he was fully alive to the practical aspects of digital computing and his 1946 report gave a masterly survey of its potential. There is no doubt that Turing saw himself in 1946 as effecting a practical implementation of his 1936 universal machine, emphasising its wide scope and ability to take on new tasks just by loading new paperwork. In (Turing 1947), a lecture to the London Mathematical Society, he gave an elegant account of the universal machine concept and in (Turing 1948) he used the term 'Practical Universal Computing Machine' to characterize all the types of digital computer then in construction. Yet those who wish to promote Turing's claim as the father of the computer, must face the difficulty that Turing himself made little of his paternity. In his 1946 report, Turing advocated the reading of von Neumann's recent EDVAC report, but did not cite his own *On computable numbers.* Even when speaking to mathematicians, in 1947, he made only brief allusion to the roots of the digital computer in mathematical logic. The 1946 report, 1947 lecture, and 1948 report—which all went unpublished—omitted to emphasize the principle of the modifiable internally stored program, with data and instructions coded alike.

That a program is itself a form of data is the basis of modern computing. We can now hardly imagine a world in which this idea was not familiar, and it is hard to read *On computable numbers* without translating Turing machines into programs and the universal machine into a computer running those programs. Turing's 1936 construction of the universal machine, by treating programs as data to be read in just the same way as numerical or any other kind of symbolic data, defined a structure quite different from that of Babbage and his successors up to the ENIAC. Yet Turing himself never identified this as the key conceptual advance, let alone claimed it for himself.

This omission is striking, as Turing certainly saw from the very start that programs were data and could be manipulated as such. On the first page of his 1946 report, Turing explained that the management of a calculation would be 'looked after by the machine itself', and this key reflexive word 'itself' goes back to Gödel's insights; so did his

proposals for having the machine compile its own machine code from instructions written in a more user-friendly form. In this plan he gave a programmed version of conditional branching, which he described by saying one could 'pretend that the instructions were really numbers'. Such 'pretence' lay at the heart of Turing's concept, coming straight from the 'description numbers' of 1936. But Turing did not expose it as such. Turing also wrote in a most important passage that the future of software writing should be free from drudgery because 'any processes that are quite mechanical may be turned over to the machine computer itself.' This insight (going far beyond von Neumann) depended on his perception of programs being themselves data-structures. In (Turing 1947) he spoke of how a machine could be given instructions in 'any symbolic logic'—yet never expanded on what he himself had done in symbolic logic. In (Turing 1948), Turing began a discussion of program modification, but cut it short with a curt reference to the constitution of the United States—presumably meaning that the constitution contained provisions for its own amendment—falling short of serious analysis. Thirty years later, Douglas Hofstadter carried forward this fascination with Gödelian self-reference at all levels, but in 1948 Turing let his insights go largely unrecorded.

Most oddly, Turing (1950) asserted that Babbage's Analytical Engine 'had all the essential ideas' of a universal machine. Yet a central argument of Turing in that paper was that programs could be modified into forms unforeseen by the original programmer: this was the thrust of his argument against 'Lady Lovelace's objection' to machine intelligence. Such program modification requires just that concept of program-as-data which Babbage's conception lacked. More generally, after moving to Manchester in 1948, Turing failed to explain and explore the intellectual history of the computer. The first semi-popular account of digital computing in Britain, *Faster than thought* (Bowden 1953), put Babbage in the prime position, and gave Turing's work a 'glossary' entry thus:

> Türing machines: In 1936 Dr Turing wrote a paper on the design and limitations of computing machines. For this reason they are sometimes known by his name. The umlaut is an unearned and undesirable addition, due, presumably to an impression that anything so incomprehensible must be Teutonic.

Turing made no counter-attack with the definitive book on The Theory and Practice of Computing that he alone could have written, and which would have completely changed the perceived history of the subject. He made no effort to promote Turing machines in the new small but growing world of computer science. At the inaugural conference for the Manchester computer he gave an absurdly low-level talk on machine code—'local conventions', not deep principles (Turing 1951b). A similar missed opportunity occurs in (Turing 1953), his paper on the calculation of the Riemann zeta-function. Even though writing for mathematicians who would have appreciated an insider's serious description of the mathematical background to computing, he listed ephemeral machine-code trivia. In his semi-popular article (Turing 1954a) he gave a serious account of computability in pure mathematics, but even there the material was self-effacing.

The one area where he certainly put himself forward, was in the philosophy of Artificial Intelligence. But this, ironically, established his reputation as a pure theorist and commentator on computing, never giving the impression that digital computers owed their principle to his logical ideas. Meanwhile, after 1950 Turing turned predominantly to his work in mathematical biology, apparently content to use (in a rather low-level manner) the universal machine rather than to explain and further develop it. All this was, of course, cut off by his death in June 1954. In his last years, however, he gave new attention to questions in *physics* which went back to concerns pre-dating his acquaintance with logic. As put by Newman (1955), Turing was 'at heart more an applied than a pure mathematician',[5] and his intellectual life had begun when his grandfather gave him a copy of Einstein's book on the theory of relativity. His first encounter with von Neumann was nothing to do with computers; it was through reading, in 1932, von Neumann's mathematical formalization of the foundations of quantum mechanics. Mathematical physics leads us back to the question of the relation of Turing's logical ideas with physical embodiment.

3 Effective calculation

In 1937, introducing Turing's ideas to logicians in the then new Journal of Symbolic Logic, Church gave his own version of Turing's definition of computability, endorsing it as a proper and valuable elucidation of the concept of effective calculability as follows:

> The author [Turing] proposes as a criterion that an infinite sequence of digits 0 and 1 be "computable" that it shall be possible to devise a computing machine, occupying a finite space and with working parts of finite size, which will write down the sequence to any number of terms if allowed to run for a sufficiently long time. As a matter of convenience, certain further restrictions are imposed in the character of the machine, but these are of such a nature as obviously to cause no loss of generality—in particular, a human calculator, provided with pencil and paper and explicit instructions, can be regarded as a kind of Turing machine.[6]

Church's summary was curiously inaccurate. Although Turing (1936a) had indeed stated at the outset that according to his definition, 'a number is computable if its decimal can be written down by a machine', he had not said anything about the finite space or

[5] A recent acquisition of the Turing Archive at King's College, Cambridge, gives further illustration of Turing's underlying knowledge and interest in mathematical physics: it is work he did in 1953-4 on an engineering problem, 'Minimum Weight Structures,' solved by going back to the principles of Lagrangian mechanics.

[6] By 'computing machine' Church could not possibly have meant his readers to understand 'Turing machine' because the readers of this first review were, by definition, ignorant of Turing machines. Indeed the expression 'Turing machine' was only coined in this review. The words 'computing machine' here imply complete generality: as Church put it in his review of Post's work, which appeared on the same page of the journal, effective calculability is to be identified with computability by an *arbitrary* machine, subject only to restrictions of finiteness.

size of machines. He had modelled his construction on a *human being* when following a rule—both the human working from explicit written instructions, and the more audacious argument from 'states of mind', which Church did not mention. His finiteness conditions were stated on this basis. Church's definition was put in terms of machines—machines with a finite *physical* characterization—and presented the human computer as a particular case. But Turing apparently made no objection and indeed Church repeated his summary in 1940, when he was more fully acquainted with Turing's work, having supervised Turing's Ph. D. thesis. In that thesis, Turing simply characterized computable operations as those 'which could be carried out by a machine', without alerting readers to any difference between his own 1936 description and Church's picture of computation by an arbitrary machine. There is no reason to suppose that Turing objected to Church's characterization of his definition, or that anyone at that time attached great importance to the distinction between human calculator and 'arbitrary machine'.

But although Turing had given a detailed and careful analysis of the concept of a human being carrying out a computation, neither Church nor Turing devoted any attention to what was involved in the concept of a machine 'occupying a finite space'. Turing, in particular, had a very good background in physics, and both of them, at Cambridge and Princeton, were surrounded by the world's most advanced work in quantum theory and relativity. They were well-placed to see that nineteenth-century concepts of space, time, and matter could longer be maintained, and that naive ideas of a 'machine' might be inadequate. Yet no such question seems to have arisen at that time.

Since then, many leading figures have raised just such questions about the physical embodiment of computation. Indeed, a fundamental question for modern computer science is whether Church's characterization of computability holds good for machines constructed from any kind of physical matter, working in any possible way. Thus the leading computer scientist A. C.-C. Yao (2003), in a series of review articles marking fifty years of the Association for Computing Machinery, summarized the Church-Turing thesis as asserting that 'any conceivable hardware system' would yield only the same functions as can be computed by Turing machines. Yao comments that this 'may not have been the belief of Church and Turing,' but that it has become the 'common interpretation.' Judging by Church's 1937 description of computability in terms of an 'arbitrary machine', he was indeed broadly on this track. However, it should be noted that Church was careful to state conditions of finiteness, which Yao's statement omits. Such a condition is obviously necessary, for one could 'conceive of' an infinite universe containing an infinite register holding an uncomputable number, thus trivially defeating the statement. We should also note that Yao regards the Church-Turing thesis as a scientific hypothesis which may or may not be true, and requires empirical testing. It is not evident that Church or Turing had so clear a view.

In fact, we should see Turing's views not as frozen in a completed thesis but as undergoing continual evolution. The expression 'Church's thesis' was not used by anyone until 1952, and Turing never cited any formulation of it as definitive. Instead, he gave various different paraphrases of the general idea, often using rather vague expressions like 'rule of thumb.' In the late 1940s, influenced by his practical work on computing,

and looking at the question of how far computing could go, Turing wrote more on the relationship between computability and physical processes. His report 'Intelligent machinery' (Turing 1948) sketched a systematic treatment. It is notable that in this report, Turing moved seamlessly between descriptions of mechanical processes as applied by human beings and by physical objects, without drawing any distinction. Thus, when describing humanly-effected algorithms, he described them as 'paper machines'. And in a reversal of the 1936 picture, he emphasised that any calculating machine could be imitated by a human being working to a rule. The focal point of his discussion was 'man as a machine,' and the themes of human and physical computation were drawn together in the question of simulating the human brain.

In (Turing 1950), the computational simulation of the physical brain was not set out quite explicitly, but it featured implicitly in (1) his estimate of the storage capacity of the brain, pointless unless its simulation is being considered and (2) the very important discussion of the 'argument from continuity of the nervous system', which would be irrelevant if the physical nature of the brain were not a central part of his philosophy. It is important also that in both 1948 and 1950, Turing gave short shrift to what he called 'the mathematical argument', declaring uncomputable functions irrelevant. The whole point of his arguments about learning and program modification was that computable operations could be capable not only of those tasks hitherto considered 'mechanical', but of features such as 'initiative'. There is no trace of his 1938 identification of 'intuition' with uncomputable steps: as indicated above, he had made a shift of view since then. The thrust of these post-war papers, putting forward what he called a 'heretical theory' of the mind, was to put forward a wholly computable model. Penrose, summarizing Turing's position in 1950, says that 'It seems likely that he viewed physical action in general—which would include the action of a human brain—to be always reducible to some kind of Turing-machine action' (Penrose 1994, p. 21). This goes a little further than what Turing explicitly wrote in 1950, but reflects what is implicit in the entire framework. With this in mind, we can consider the more detailed statements about physical machines which Turing offered in this period, the nearest thing to the analysis of the human computer he had given in 1936.

4 Physical action, continuity and randomness

The brain was the centre of Turing's interest in machines. And as he famously remarked, the fact that the brain has the consistency of cold porridge is not central to the analysis of 'intelligent machinery'. It is not the physical mass or chemical composition of the brain that we must analyse, but its logical function. In his 1948 report, Turing made a distinction between 'active' and 'controlling' machinery, using 'a bulldozer' as an example of the former. It is controlling or information-theoretic machinery that we are concerned with. This is in one sense a trivial remark—in discussing computation we are only interested in machines *qua* calculating machines. But it does open up a deeper perspective: what physical properties must a system possess to be capable of storing and operating on information? Turing did not tackle this general question, but as we

shall see, the physical embodiment of information-theoretic machines was ever-present in his discussion.

In particular, he wished to relate the *continuous* world of physics and the *discrete* world of Turing machines. In this 1948 report he explicitly made a distinction between discrete and continuous machines. The main point of making this distinction lay in Turing's classification of the brain as a machine which is continuous, but in such a way that the continuity is not relevant to its function.

It is important to note that he was experienced in theory and practice with continuous systems and their relationship to the discrete. Turing's example of 'a telephone' to illustrate what he meant by a continuous machine hardly conveyed the depth of knowledge he brought to this analysis. Nor could he have done—official secrecy would have prevented it. In early 1943 he had privileged access, as the representative of the British government, to the most secret American research in secure speech systems. These methods used advanced sampling theory and encipherment through partly digital and partly analogue electronic techniques. From May 1943 onwards he devised and built his own speech encipherment system, 'Delilah' (Turing 1944). But this is only one of multifarious examples throughout his work. The mathematical work that attracted von Neumann's attention in 1938 was his work on the discrete approximation of continuous groups; his morphogenetic work centred on the evolution of discrete structure from non-linear partial differential equations. His 1946 computer proposals were in no way limited to the discrete work of a logician. He included from the start a discussion of floating-point storage, and applications to traditional applied mathematical questions such as potential theory and physical chemistry. In (Turing 1948b) he pioneered important new numerical methods in matrix inversion—for a recent detailed discussion see (Higham 1996, p. 188)—essential to the use of digital computing in serious physical and engineering problems. Most importantly, his most mathematical lecture (Turing 1947) began by advancing the view that discrete computation offered an improvement on analogue machines such as differential analysers. In an argument based on the principle of the universal machine, he explained that instead of building bigger or better analogue machines, one could gain increased accuracy, without limit, by the use of programming and greater storage capacity alone. Very considerable effort had been put into devising differential analysers by leading figures in mathematical physics, and in short, Turing had to know what he was talking about.

So it is worthwhile to note what Turing (1950) gave as the reason why the brain was effectively discrete. He did not mention the features of neurons that had inspired McCulloch and Pitts to model them with logical gates. Instead, he started with a statement that would apply to all non-trivial continuous dynamical systems:

> The displacement of a single electron by a billionth of a centimetre at one moment might make the difference between a man being killed by an avalanche a year later, or escaping. It is an essential property of the mechanical systems which we have called discrete state machines that this phenomenon does not occur.

Out of all his knowledge in mathematics, physics and engineering, this phenomenon—

often called the 'butterfly effect'—was the aspect of the relation between **Z** and **R** that he chose to emphasise. In this paper, Turing apparently took for granted the argument for approximation by use of finite-difference methods such as he had explained in (Turing 1947). Instead, he jumped ahead to answer an objection which few of his readers in 1950 would have thought of posing: that such approximation fails when chaotic effects dominate. This insight could have be used as an important opening into the modern study of non-linear dynamical systems. But that was not how he treated it. In the context of computing, he held such chaotic phenomena to be irrelevances, of no functional significance at all.[7]

Turing explained that the continuity of the brain meant that it would have such 'avalanche' effects, which a discrete system would not. 'A small error in the information about the size of a nervous impulse impinging on a neuron, may make a large difference to the size of the outgoing impulse.' But, he stated, the functional effect could be imitated by a discrete system if *random choices* were added to the actions of the discrete machine. The argument was sketchy and he referred to randomness very vaguely, saying that the digits of π would do as a source of random input. It is mildly surprising that he did not take more interest in defining and analysing randomness, considering his long-standing interest in the analysis of probability and statistics, including the study of normal numbers we have already noted, and the fact that his war work had largely turned on the problem of distinguishing pseudo-random from random. This is a subject he might well have taken up later, had he lived. A late-1950s Turing might also, in the light of his work in morphogenesis, seeing discrete structures emerging out of non-linear equations, have taken more interest in exploring the relationship of computability with continuous dynamical systems. There is now great interest in theories of computable analysis and analogue computing which seek to address this gap. But in 1950 at least, Turing seems to have considered it quite sufficient to consider computation as a discrete process, with the addition of rather cursorily considered random elements.

We may contrast Turing's approach with that of some modern theorists, who seek to outdo discrete computation by exploiting the very elements that Turing made little of. Thus Siegelmann (1999) has suggested 'computing beyond the Turing limit' by harnessing a system with an uncomputable element of **R** as a physical parameter. In her construction, a precise knowledge of that parameter is vital. Such proposals, however, have an infinite difficulty attached to them, which follows simply from the nature of real numbers as equivalent to an infinite set of discrete data. For uncomputable numbers there is no possibility of expressing this infinite set in terms of a finite package of data.

The problem is that every uncomputable number can be approximated arbitrarily closely by a computable number, and indeed by a rational number. Thus infinite precision is needed to distinguish the computable from the uncomputable. Yet gravitational waves from a neutrino collision in a faraway galaxy will wipe out all but the first few decimal places of any physical measurement, and this would seem to make an uncomputable parameter, even if it could be supplied, impossible to exploit. As a general rule,

[7] Penrose (1989, p. 173) gives an argument to the same effect, that chaotic dynamics are 'useless' to the brain.

real numbers are used in physics to express quantities in such a way that the successive digits can be assumed to become less and less significant. But if the *uncomputability* of a real number is the object of attention, rather than its magnitude, this assumption fails, for uncomputability lies in that infinite tail of 'least significant' digits. This contradicts the standard idea of a mathematical model of a physical system, which is meaningless without stability and robustness in the face of infinitesimal perturbations.

A similar difficulty faces those who appeal to the theorem of a group including Shannon (De Leeuw et al. 1956), which concerns the possible outputs of a Turing machine with a random element—something that Turing, with his vague definition of randomness, had left open. Their result states that if the parameter of the element is α, such a machine can compute everything that is computable relative to α. Hence if α is an uncomputable real number, such a machine can apparently outdo Turing computability. But again, it is necessary for this that α is embodied in the system with infinite precision, and this is as unphysical a demand for a probabilistic parameter as for any other. A related idea, also currently popular, is that it should be possible to exploit the fact that a random real number is, with probability 1, uncomputable. Again, a difficulty is that it must be known as an infinite sequence for its uncomputability to play any role. But there is a further difficulty for those who hope to use this property to transcend the limitations of computability: how can adding randomness into a computation possibly *increase* its predictive power? A confusion may perhaps have arisen from Chaitin's very interesting characterization of numbers like his Ω as random, in the sense that they are totally patternless and unpredictable. Every 0-or-1 bit in Ω means something very non-random, e.g. the truth of the Fermat-Wiles theorem. This is quite the opposite situation to that of a number produced by a random process, where no 0-or-1 bit has any significance. Thus using Ω as an oracle indeed is like being able to draw on infinite wisdom, but having access to a random number gives no useful knowledge at all.

Turing never made any such suggestion about using random numbers as infinite sources of information, and his discussion could hardly be more different from infinite-precision proposals. In fact, Turing emphasised the error inevitable even in systems demanding only finite precision. That is, he asserted the physical impossibility of embodying even computable processes. His 1946 computer design showed his keen awareness of hardware error problems, and in (Turing 1947) he gave a brief discussion of how discrete states are identifiable with discrete regions of the configuration space of a dynamical system—thus making allowance for the vagaries of a continuous hardware system. But Turing noted that computations would still be subject to error when particularly large thermodynamic fluctuations occurred. In 1948 he came up with an estimate that there was virtual certainty of error in $10^{10^{17}}$ years.

But we now turn towards a more surprising observation by Turing about physics and computation. In (Turing 1948) he also pointed out the constraint on computation imposed by the finite speed of light. We may be amused now by his assumption that components would need a centimetre of spatial separation, but he was perfectly correct in drawing attention to the physical fact which drives miniaturisation. Thus Turing acknowledged the implications for computing of the basic reality of relativistic space-

time. What about quantum mechanics, the other great twentieth-century development in physical theory? In 1948 Turing made no reference to the limits imposed on computing by the quantum-mechanical nature of the electron. In particular, he did not suggest anything like the modern theory of quantum computing. But he did see another place in which quantum mechanics might be important.

It came in 1951, in Turing's radio talk 'Can a digital computer think?' (Turing 1951a). Here Turing mostly went over the content of his 1950 paper, but with a more explicit statement of the problem of simulating on a computer the action of the brain, thought of as a physical machine. When doing so, he added a quite new comment: such simulation would only be possible for machines

> ... of the sort whose behaviour is in principle predictable by calculation. We certainly do not know how any such calculation should be done, and it has even been argued by Sir Arthur Eddington that on account of the Indeterminacy Principle in Quantum Mechanics no such prediction is even theoretically possible.

This sentence marks a change of view from 1950. He had made no mention then of this problem when discussing the 'continuity of the nervous system', and the force of his discussion was that any such effects could be simulated satisfactorily by random elements. Turing only wrote this one sentence on the relationship of computation to quantum mechanics, so we know no more of why he had decided that physics now had to be taken more seriously. It should be noted that Turing does not actually state a view of his own; he attributes the objection to Eddington and presents it as almost fanciful. But we do know that Turing then went on to spend time trying to formulate 'a new quantum mechanics'. Most probably he was hoping to close this gap in computability, by finding a response to this 'Eddington argument'. But we cannot tell; his ideas might have led in new directions. We do know that he focussed on the principle of wave-function 'reduction' that underlies the indeterminacy or uncertainty principle. In his last months he came up with the 'Turing Paradox' (Gandy 1954)—that the standard axioms of quantum mechanics imply that in the limit of continuous observation a quantum system cannot evolve.

Turing's reference to Eddington is particularly striking: it was from *The nature of the physical world* (Eddington 1928) that Turing first learnt of quantum mechanics. Eddington clearly influenced Turing's youthful view that quantum-mechanical uncertainty explained the apparent free will of the mind, rescuing it from Laplacian determinism (Turing 1932?).

5 Beyond Turing machines?

The word 'hypercomputation' has been coined to refer to processes supposed to be capable, within a finite time, of effecting a procedure that no Turing machine could carry out. There are many attempts to devise such processes and so defeat Church's

characterization of computability. For example, a simple observation is that an 'accelerated' Turing machine can solve a classically unsolvable problem in a finite time, if its steps get faster and faster without limit as the computation proceeds. By summing an elementary geometric series, infinitely many steps can be taken in a finite total time. This is, however, a very large—indeed infinite—'if'. It is easy to see that were such a machine to exist, then given any time $\epsilon > 0$, any computation, or indeed an infinite sequence of computations, could be done in time less than ϵ. The unbounded smallness of ϵ is essential to the nature of uncomputability. The bland term 'accelerated' for this imaginary concept is an infinite understatement of the completely different physical world that would be required for it to exist.

Such a construct trades the infinite *time* required by a Turing machine computation for the unbounded *speed* of this imaginary entity. A similar trade-off can be applied by postulating that an infinite quantity of data can be stored in a finite space, and similar difficulties ensue. Stannett (2004), for instance, suggests the encoding of infinite data in a wire, by making use of the Fourier decomposition of its shape. Stannett concedes that this procedure 'may' require wavelengths going below atomic size, and would 'probably not be possible using current technology', but suggests using sound waves instead, referring to wavelengths measured in 'thousandths of a metre'. This is again an infinite understatement of the proposition. Given any length $\epsilon > 0$, *almost all* the information must be encoded in wavelengths of less than ϵ. Somehow the bounds imposed by the atomic nature of matter, and indeed the Planck length, would have to be magically suspended. If the physical basis for such unboundedly high-pitched sounds existed, it is hard to see why one should stop at storing one uncomputable number; one might as well store infinitely many, and by the rules of Hilbert's Hotel, there would be plenty of room for infinitely more. This would indeed revolutionise computer storage.[8]

In contrast, the very interesting models of Warren D. Smith, cited by Yao (2003) as important discussions of computability in physical theory, are highly non-trivial in their mathematical analysis. They shed interesting new light on the non-linear equations of physics. But, as Smith makes clear, and as Yao emphasises, the results depend on infinite precision, typically assuming that equations such as the Navier-Stokes hold at all length-scales and so ignoring the atomic basis of matter. In other proposals the infinite resources demanded may be less clearly stated. The proposal of Kieu (2004, 2005) for quantum computing, for instance, requires infinite precision and data density, but the advanced nature of the material (and its confused argument) makes this opaque (Smith 2006, Hodges 2005). Kieu's proposal is, incidentally, an example of a so-called 'stochastic' decision procedure, meaning that it offers only a certain probability of reaching a correct decision. Thus an infinite time would still be taken to reach certainty, and it is not obvious why this infinitude should be forgiven while the infinite time taken by a Turing machine to solve the halting problem is not.

A very reasonable approach would be to extend or refine Church's finiteness condi-

[8]It is of course questionable whether an infinite lookup table, even if such a thing existed, could really be said to be 'solving' a problem. There is a serious problem concerning how such an oracle could ever be verified, and therefore of why it should ever be believed.

tion in such a way as to exclude such trade-offs which merely convert infinite time or infinitely large data registers into requirements for unbounded speed, precision or data density. Martin Davis, by asserting that 'all that we experience is finite', and ruling out 'hypercomputation' on those grounds, takes such an approach (Davis 2006). But experience does not tell us the fundamental physics of space, time and matter; indeed science still lacks a complete theory. A viewpoint closer to Turing's would be that discoveries in physics and logic may influence each other, allowing understanding of the Church-Turing thesis to evolve in unforeseen directions. There are already developments in quantum computing which indicate the need for such evolution. Here are two: (1) Quantum gates need to be understood as unitary operations, based on continuous unitary evolution in a continuum of physical states. In an extreme case, the 'adiabatic cooling' method reduces the whole of a computation to a single gate. This runs against the traditional assumption of logicians that an effective calculation must be performed as a sequence of discrete steps. (2) Quantum 'teleporting' depends on quantum entanglement, wave-function reduction, and space-time structure, in a way that is not wholly understood. While at school, Turing (1929) commented on entanglement: 'Of course [Schrödinger] does not believe that there are really about 10^{70} dimensions, but that this theory will explain the behaviour of an electron.' The ontology of quantum field theory, and so of computations done with it, is hardly better understood now than it was by the seventeen-year-old Alan Turing.

The concept of 'finite data' in physics is not as simple as the structure of $\mathbf{Z}$, because 'quantum information' involves wave-function and space-time geometry in a manner that defeats our intuition. In a recent comment (Hodges 2004) on what Turing might have done had he lived, attention was drawn to the counterfactual 'bomb-testing' use of the quantum reduction process (Elizur and Vaidmann 1993), and since then there has been more work on such 'interaction-free measurement', some bringing in what is now called the 'quantum Zeno effect'—which is just the prediction that Turing noted in 1954. Overall, it would appear that the physical words in Church's formulation of computability in 1937—'finite space' and 'sufficiently long time' —point correctly to what is necessary. We still need a full and fundamental understanding of space-time—or more likely, of something *beyond* our current space-time picture, e.g. a super-string or twistor geometry. Penrose (2004) gives a recent discussion of the quest for such deeper structure, such as theoretical physicists now confidently expect to find.[9]

It would be unfortunate if rather trivial 'hypercomputation' schemes requiring unbounded speed or data density should draw attention away from the genuinely interesting new models of computation arising from twentieth-century physics. Another unfortunate aspect of the word 'hypercomputation', to which Martin Davis has drawn attention, is that the term imparts a positive connotation of effective engineering. It suggests the ability to pose a problem, classically unsolvable, to a hypercomputer which will correctly answer it. Yet the term is used in contexts where it means nothing but the negative and purely mathematical word 'uncomputable'. Thus Stannett (2004) counts as 'a hyper-

[9]Thus, in the treatment of the 'gluon' field holding the proton together, it can now be seen as very helpful to frame the work in a twistor space, a complex-analytic extension of space-time (Hodges 2006a).

computational model' such things as Turing's purely mathematical structures of ordinal logics. Stannett also lists Penrose's theory as such a 'model', but this overlooks a more subtle difference. Penrose indeed believes that the physical function of the brain requires some law of physics, as yet unknown, which cannot be approximated by a computable function. He believes that the quantum-mechanical reduction process is not in fact random but governed by some quite unknown law. It is striking that Penrose, like Turing, is drawn to the physics of the brain and the question of its quantum-mechanical basis, and sees the possibility of an element in wave-function reduction which cannot be simulated by a computable process. But he does not propose building a machine for performing a hypercomputation. Penrose emphasises that the unknown physics must be 'something completely different' from an oracle supplying an uncomputable function.

Prominent proponents of 'hypercomputation' currently advise the general public of a possible, even imminent revolution in technology. The largely crude and unphysical manifestos for this revolution make unconvincing economic forecasts. On the other hand, there may be something very much more subtle yet to be discovered about the fundamentals of physics, such as is suggested by Penrose's theory of wave-function reduction, which will require a new logic of information, communication and computation and perhaps indeed new insights into discrete and continuous structure. If so, Turing's return to mathematical physics was a good pointer to the way forward. In this paper and in some other recent contributions (Hodges 2004, 2006b, 2007a, 2007b), the present author urges that logic and physics should not be seen as wholly disjoint activities.

In looking back fifty years to Alan Turing, we can do well to look back a hundred years to the birth of the Modern: the era of Planck's constant, Hilbert's problems, Russell's paradoxes and Einstein's relativity. At the centre of the century, which Turing did so much to shape, it was physics that took the front of the stage, with cryptology and computers in comparative obscurity. Alan Turing's life story was part of the secrecy and obscurity of the logical world. Only some thirty years after his death did common culture appreciate that computation might seriously impinge on human life. He was a generation ahead, and it needed a generational shift to appreciate the issues of his life and death. Only with the lifting of an immense secrecy could Alan Turing's individualism be seen as being both, in June 1944, vital to Anglo-American security, and in June 1954, a threat to it.

Now, when computing is front-page news, physics has lost its 1950s status. But computing requires both software and hardware, and the Church-Turing thesis, which underlies computing, relates both to logic and to physics. What Turing analysed to such great effect in 1936 was the idea of doing finitely many things at a time—but those words 'doing', 'finite', 'thing', 'at', 'a', 'time', apparently so simple, are far from clear at the level of fundamental physics. It is not just the relationship between the integers and the reals that is involved: far more subtle features of the complex numbers and analytic functions seem to enter in an essential way into physical reality, for reasons which are far from clear. Alan Turing's last postcard (Turing 1954b)—which should not be taken as the thesis of any church!—is an appropriate ending. Or perhaps it is a starting-point for the next fifty years:—

Hyperboloids of wondrous Light
Rolling for aye through Space and Time
Harbour those waves that somehow Might
Play out God's holy pantomime.

6 References

V. Bowden (ed.) (1953), *Faster than Thought,* (Pitman, London, 1953)

J. L. Britton (1992), Introduction and notes to Turing's note on normal numbers, in *The collected works of A. M. Turing: Pure mathematics*, ed. J. L. Britton (North-Holland, 1992)

D. G. Champernowne (1933), The construction of decimals normal in the scale of ten, J. Lond. Math. Soc. **8** 254-260

G. Chaitin (2006), The limits of reason, *Scientific American*, March 2006

A. Church (1937), Review of Turing (1936), J. Symbolic Logic **2**, 42-3

M. Davis (2006), Why there is no such discipline as hypercomputation, to appear in Applied Math. Comp.

K. De Leeuw, E. F. Moore, C. E. Shannon and N. Shapiro (1956), Computability by probabilistic machines, in *Automata Studies*, eds. C. E. Shannon and J. McCarthy (Princeton University Press, 1956)

A. S. Eddington (1928), *The nature of the physical world,* (Cambridge University Press, 1928)

A. C. Elizur and L. Vaidman (1993), Quantum-mechanical interaction-free measurements, Found. of Physics **23**, 987-997

R. O. Gandy (1954), letter to M. H. A. Newman, available at www.turingarchive.org, item D/4. Text in *The collected works of A. M. Turing: Mathematical Logic.* eds. R. O. Gandy and C. E. M. Yates (North-Holland, 2001)

I. J. Good (1992), Introductory remarks for the article in *Biometrika* **66** (1979), 'A. M. Turing's statistical work in World War II', in *The collected works of A. M. Turing: Pure mathematics*, ed. J. L. Britton (North-Holland, 1992)

I. J. Good (1993), Enigma and Fish, in *Codebreakers*, eds. F. H. Hinsley and A. Stripp (Oxford University Press, 1993)

I. J. Good (2001), Commentary on Turing's manuscript 'Minimum cost sequential analysis', in *The collected works of A. M. Turing: Mathematical Logic*, eds. R. O. Gandy and C. E. M. Yates (North-Holland, 2001)

N. J. Higham (1996), *Accuracy and stability of numerical algorithms,* (SIAM, 1996)

A. Hodges (1983), Alan Turing: the enigma (Burnett, London; Simon & Schuster, New York; new edition Vintage, London, 1992)

A. Hodges (2004), What would Alan Turing have done after 1954, in *Alan Turing: Life and legacy of a great thinker,* ed. C. Teuscher (Springer, 2004)

A. Hodges (2005), Can quantum computing solve classically unsolvable problems?, http://arxiv.org/abs/quant-ph/0512248

A. Hodges (2006a), Scattering amplitudes for eight gauge fields, http://arxiv.org/abs/hep-th/0603101

A. Hodges (2006b), The essential Turing, book review, Notices of the A. M. S. **53**, 1190-1199, available as http://www.ams.org/notices/200610/rev-hodges.pdf

A. Hodges (2007a), What did Alan Turing mean by 'machine', in *The mechanical mind in history,* eds. P. Husbands and O. Holland, M. Wheeler, (MIT Press, 2007).

A. Hodges (2007b), Alan Turing, logical and physical, in *New computational paradigms,* eds. B. Cooper, B. Löwe, A. Sorbi (Springer, 2007)

T. D. Kieu (2004): An anatomy of a quantum adiabatic algorithm that transcends the Turing computability, http://arxiv.org/quant-ph/0407090,

T. D. Kieu (2005): Hypercomputability of quantum adiabatic processes: facts versus prejudices, http://arxiv.org/quant-ph/0504101

K. Knopp (1921), *Theory and application of infinite series* (original German edition 1921; English edition: Blackie, 1928)

R. Monk (1996), *Bertrand Russell: the spirit of solitude* (Jonathan Cape, 1996)

M. H. A. Newman (1955), Alan Mathison Turing, Biographical memoirs of Fellows of the Royal Society **1**, 253-263

R. Penrose (1989), *The emperor's new mind* (Oxford University Press, 1989)

R. Penrose (1994), *Shadows of the mind* (Oxford University Press, 1994)

R. Penrose (2004), *The road to reality* (Jonathan Cape, 2004)

H. T. Siegelmann (1999), Neural networks and analog computation: beyond the Turing limit (Birkhäuser, Boston, 1999)

W. D. Smith (2006), Three counterexamples refuting Kieu's plan for "quantum adiabatic hypercomputation"; and some uncomputable quantum mechanical tasks, to appear in Applied Math. Comp., available also at http://www.math.temple.edu/∼wds/homepage/works.html

M. Stannett (2004), Hypercomputational models, in *Alan Turing: Life and legacy of a great thinker,* ed. C. Teuscher (Springer, 2004)

A. M. Turing (1929) Letter to his mother dated 20 November 1929, available at www.turingarchive.org, item K/1. Text in (Hodges 1983, p. 40)

A. M. Turing (1932?), Handwritten essay: Nature of Spirit. Photocopy available in www.turingarchive.org, item C/29. Text in (Hodges 1983, p. 63)

A. M. Turing, (1936a), On computable numbers, with an application to the Entscheidungsproblem, Proc. Lond. Math. Soc. (2) **42**, 230-265

A. M. Turing (1936?b), A note on normal numbers, manuscript and typescript available at www.turingarchive.org, item C/15. Text in *The collected works of A. M. Turing: Pure mathematics*, ed. J. L. Britton (North-Holland, 1992)

A. M. Turing (1937), On computable numbers, with an application to the Entscheidungsproblem. A correction, Proc. Lond. Math. Soc. (2) **43**, 544-546

A. M. Turing (1938a), Ph. D. thesis, Princeton University, published as: Systems of logic based on ordinals, Proc. Lond. Math. Soc. (2) **45**, 161-228 (1939)

A. M. Turing (1938b), Finite approximations to Lie groups, Ann. of Math. **39** (1), 105-111

A. M. Turing (1938c), The extensions of a group, Compositio Math. **5**, 357-367

A. M. Turing (1939), Blueprint of machine, prepared with D. C. McPhail, available at www.turingarchive.org, item C/2

A. M. Turing (1943), A method for the calculation of the zeta-function, Proc. Lond. Math. Soc. (2) **48** 180-197

A. M. Turing (1944), Speech system 'Delilah' – report on progress, typescript dated 6 June 1944. I am indebted to Ralph Erskine for telling me of this report in the National Archives, box HW 62/6

A. M. Turing (1946), Proposed electronic calculator, copy of typescript available at www.turingarchive.org, item C/32. Text published in various forms, e.g. in *The collected works of A. M. Turing: Mechanical Intelligence*, ed. D. C. Ince (North-Holland, 1992)

A. M. Turing (1947), Lecture to the London Mathematical Society, 20 February 1947, typescript available at www.turingarchive.org, item B/1. Text published in various forms, e.g. in *The collected works of A. M. Turing: Mechanical Intelligence*, ed. D. C. Ince (North-Holland, 1992)

A. M. Turing (1948a), Intelligent machinery, National Physical Laboratory report, typescript available at www.turingarchive.org, item C/11. Text published in various forms, e.g. *The essential Turing.* ed. B. J. Copeland (Oxford University Press, 2004)

A. M. Turing (1948b), Rounding-off errors in matrix processes, Quart. J. Mech. Appl. Math. **1**, 287-308

A. M. Turing (1950), Computing machinery and intelligence, Mind **59**, 433-460

A. M. Turing (1951a) Can digital computers think? BBC talk, typescript available at www.turingarchive.org, item B/5. Text published in *The essential Turing.* ed. B. J. Copeland (Oxford University Press, 2004). Copeland drew attention to the quoted passage in an earlier (1999) edition.

A. M. Turing (1951b) Local programming methods and conventions, in the Proceedings of the Manchester University Computer Inaugural Conference. Reproduced in *The collected works of A. M. Turing: Mathematical Logic.* eds. R. O. Gandy and C. E. M. Yates (North-Holland, 2001), p. 249, with a commentary by M. Campbell-Kelly.

A. M. Turing (1953), Some calculations of the Riemann zeta-function, Proc. Lond. Math. Soc. (3) **48**, 99-117

A. M. Turing (1953-4), Minimum weight structures, typescript with introduction by R. K. Livesey, available at www.turingarchive.org, item C/33

A. M. Turing (1954a), Solvable and unsolvable problems, Science News **31**, 7-23

A. M. Turing (1954b), Postcards to Robin Gandy, available at www.turingarchive.org, item D/4. Text in *The collected works of A. M. Turing: Mathematical Logic.* eds. R. O. Gandy and C. E. M. Yates (North-Holland, 2001)

A. C.-C. Yao (2003), Classical physics and the Church-Turing thesis, Journal of the ACM **50**, 100-105