

# A Brief History of Algebra and Computing: An Eclectic Oxonian View

Jonathan P. Bowen

Oxford University Computing Laboratory  
Wolfson Building, Parks Road, Oxford OX1 3QD

Dedicated to Prof. C.A.R. Hoare, FRS, James Martin Professor of Computing at the  
Oxford University Computing Laboratory.

Completed on his 60th birthday, 11th January 1994.

*That excellent woman knew no more about Homer than she did about Algebra, but she was quite contented with Pen's arrangements . . . and felt perfectly confident that her dear boy would get the place which he merited.*

Pendennis (1848–50), by William Makepeace Thackeray (1811–1863)  
(*The story of the progress of an Oxford student.*)

**Oh heck—another hour of algebra!**

$$\left( \frac{\mathbf{O}pposite\ side}{\mathbf{H}ypoteneuse} = \text{sine}; \frac{\mathbf{A}djacent\ side}{\mathbf{H}ypoteneuse} = \text{cosine}; \frac{\mathbf{O}pposite\ side}{\mathbf{A}djacent\ side} = \text{tangent} \right)$$

## 1 The Origins of Algebra

The following problem on the Rhind Papyrus in the British Museum, London, was written in about 1650 BC:

Divide 100 loaves among 10 men including a boatman, a foreman and a door-keeper, who receive double portions. What is the share of each?

This can of course be solved using algebra.

The first treatise on algebra was written by Diophantus of Alexandria in the 3rd century AD. The term derives from the Arabic *al-jabr* or literally “the reunion of broken parts.” As well as its mathematical meaning, the word also means the surgical treatment of fractures. It gained widespread use through the title of a book *ilm al-jabr wa'l-mukabala*—the science of restoring what is missing and equating like with like—written by the mathematician Abu Ja'far Muhammad (active c800–847), who subsequently has become known as *al-Kharazmi*,

the man of *Kwarazm* (now Khiva in Uzbekistan). He introduced the writing down of calculations in place of using an abacus. *Algorism* (the Arabic or decimal system of writing numbers) and *algorithm* both derive from his name.

Algebra was brought from ancient Babylon, Egypt and India to Europe via Italy by the Arabs.

## 2 Early English Algebra

In the first half of the 16th century, Cuthbert Tonstall (1474–1559) and Robert Recorde (1510?–1558) were two of the foremost English mathematicians [2]. They were the first mathematicians at the University of Cambridge whose lives have been recorded in any detail and as such may be considered founders of one of the most important centres of mathematics in the world. Both migrated to Oxford University during their careers. Robert Recorde, perhaps the more important of the two, became a Fellow of All Souls College at Oxford in 1531.

The earliest use of the word *algebra* may be found in Recorde's *Pathway of Knowledge* (1551) in which he wrote:

Also the rule of false position, with dyvers examples not onely vulgar, but some appertayning to the rule of Algebra.

In 1557 he introduced the equality sign '=' in his *Whetstone of Witte*, chosen "because noe 2 thynges can be moare equalle" (than two parallel lines of the same length). The symbols '+' and '-' were introduced for the first time in print in John Widmann's *Arithmetic* (Leipzig, 1489), but only came into general use in England after Recorde's *Whetstone*:

There be other 2 signes in often use of which the first is made thus -+- and betokeneth more: the other is thus made — and betokeneth lesse.

These symbols may have originated from marks chalked on chests of merchandise in German warehouses, indicating the variation from some standard weight. They appeared in a work by Stifel in 1544.

Thomas Hariot (1560–1621), a native of Oxford, at St. Mary Hall (subsequently incorporated with Oriel College), invented the signs of inequality '<' and '>' [3]. He continued the work of Recorde, helping to give algebra its modern form. His work on algebra was published ten years after his death, although it was probably written around 1610, including Hariot's law of Signs concerning roots.

William Oughtred (1574–1660), a private tutor to Oxford students, worked on mathematics at a country vicarage and extended the use of the multiplication sign '×' in his *Clavis Mathematica* (1631), previously used in a rather different manner by Recorde in his *Grounde of Artes* (1542). He also introduced the trigonometrical terms *sine*, *cosine* and *tangent*. The division sign '÷' was used by J. H. Rahn in 1659 and was introduced into England by John Pell in 1668.

### 3 Algebra and Analytical Engines

In 1827, John Hershel wrote enquiring whether Charles Babbage (1791–1871) [8] wished to be considered for the Savilian chair of mathematics at Oxford University, which was then vacant. Babbage’s father had just died and on 12th February Georgina Babbage, his wife, replied saying:

It is seldom that I cannot say a priori what Charles *would* like or would not like, but with regard to this professorship I cannot judge. Before his father’s death he would have taken it, but whether this event wil make him *more* or *less* desirous of obtaining it, without more knowledge of our affairs I cannot say.

In the event, the appointment of the professorship had already effectively been decided. Babbage did not apply and Baden Powell was offered the Chair. Instead Babbage became Lucasian Professor of Mathematics at Cambridge, a post previously occupied by Sir Isaac Newton. He was influential in introducing the continental European notations of the German mathematician Gottfried Wilhelm Leibniz (1646–1716) into Britain. The Newtonian notation prevalent until then had been detrimental to the progress of mathematics compared to the rest of Europe (e.g., the use of the Newtonian ‘dot’ notation for differential calculus instead of the more powerful Leibniz ‘d’s notation). But if Cambridge mathematics was in a poor state at the time despite its renown, Oxford mathematics was even worse!

By 1836, Babbage was already considering the design for an Algebra or Formula Engine, a generalised version of his more famous Analytical Engines. Ada Lovelace’s notes for the paper by Luigi Federico Menabrea on Babbage’s work state:

Many persons who are not conversant with mathematical studies, imagine that because the business of the engine is to give its results in *numerical notation*, the *nature of its processes* must consequently be *arithmetical* and *numerical*, rather than *algebraic* and *analytic*. This is an error . . . . The engine might develop three sets of results . . . *symbolic* results . . . ; *numerical* results . . . ; and *algebraical* results in literal notation.

The potential symbolic results of Babbage’s Analytical Engines are important in connection with their relation to modern computers. Ada Lovelace noted that:

. . . each circle at the top [of a column of wheels recording decimal digits] is intended to contain the algebraic sign + or –, . . . In a similar manner any other purely *symbolic* results of algebraic processes might be made to appear in these circles.

On 10 July 1836, soon after Babbage had considered a punched-card input system, he recorded the following:

... this day I had a general but only indistinct conception of the possibility of making an engine work out *algebraic* developments ... I mean without *any* reference to the *value* of the letters ... My notion is that as the cards (Jacquards) of the calc. engine direct a series of operations and then recommence with the first so it might perhaps be possible to cause some of the cards to punch others equivalent to any given number of repetitions. But these hole[s] might perhaps be small pieces of formula previously made by the first cards and possibly some mode might be found of arranging such detached parts.

Next year, on 13 December 1837, Babbage noted the following in one of his sketchbooks:

On machinery for the Algebraic Development of Functions

About this date the idea of making a development engine arose with considerable distinctness. It is obvious that if the Calculating Engine could print the substitutions which it makes in an Algebraic form we should arrive at the algebraic development 'it can print all those substitutions which are noted in the *composition* in some of the notations'—it will however be better to construct a new engine for such purposes.

I.e., he was considering the construction of an engine for general algebraic operations and printing formulae resulting from calculations on such a machine. These ideas may be regarded as an extension of the Analytical Engines, but were not developed further by Babbage.

Towards the end of his life Babbage asked Harry Wilmot Buxton to write his biography. An early draft, together with other papers Babbage lent him whilst undertaking this task, are now held as part of the Buxton collection at the Museum of the History of Science in Oxford.

## 4 Boolean Algebra

Leibniz initiated the search for a system of symbols with rules of their combination in his *De Arte Combinatoria* of 1666, as well as developing the binary notation. In 1854, George Boole (1815–1864), Professor of Mathematics at Cork from 1849 despite having no first degree, formalised a set of such rules in the seminal work entitled, perhaps optimistically, *An Investigation of the Laws of Thought*. Boole's aim was to identify the rules of reasoning in a rigorous framework and revolutionised formal logic after thousands of years of little progress. They transformed logic from a philosophical into a mathematical discipline. These rules have subsequently become known as Boolean algebra and the design of all modern binary digital computers has depended on the results of this work. These logical operations, normally implemented as electronic *gates*, are all that are required to perform more complicated operations such as arithmetic.

Charles Lutwidge Dodgson (1832–1898) [1], a Mathematics Lecturer at Christ Church, Oxford from 1855 to 1881, was influenced by the work of Boole. He had a general interest in algebra and also teaching. In May 1855 he noted in his diary:

I began arranging a scheme for teaching systematically the first part of Algebraic Geometry: a thing which no one hitherto seems to have attempted—I find it exceedingly difficult to do it in anything like a satisfactory way.

He subsequently produced works on *The Fifth Book of Euclid proved Algebraically* and *A Syllabus of Plan Algebraical Geometry* as well as collections of algebraic and arithmetic formulae to aid examination candidates.

Much later, in his diaries of 1884 Dodgson noted: “In these last few days I have been working on a Logical Algebra and seem to be getting to a simpler notation than Boole’s.” In 1885 he notes: “I have occupied myself at Guildford in teaching my new ‘Logical Algebra’ to Louisa, Margaret and the two boys.” In March 1885 he mentions “A Symbolic Logic, treated by my algebraic method.” He published a number of works on logic, including *Symbolic Logic, Part I: Elementary* in 1896 under his more famous alias of Lewis Carroll. Unfortunately Part II never appeared.

## 5 Algebra and Computing

From the end of the 2nd World War in 1945 the world was set for an exponential growth in the use of computers. At this time, Leslie Fox (1918–1992) [9] moved from the Admiralty Computing Service to the National Physical Laboratory (NPL) joining a section including Alan Turing [7] and led by E. T. Goodwin. He was interested in numerical linear algebra and whilst at the NPL he started a line of investigation into using Gaussian elimination to estimate the accuracy with solving linear equations. Prof. Fox went on to become the first director of the Oxford University Computing Laboratory in 1957, becoming Professor of Numerical Analysis in 1963. He stayed there until his retirement in 1983.

J. R. Womersley was the Superintendent of the Mathematics Division at the NPL. In 1946 he noted in a report on the proposed ACE computer that:

... this device is not a calculating machine in the ordinary sense of the word. One does not need to limit its functions to arithmetic. It is just as much at home in algebra ...

In 1951, Christopher Strachey, then a teacher at Harrow School, made contact with Mike Woodger at the NPL via a mutual friend. He started to write a draughts program for the Pilot ACE, and soon progressed to the machine being developed at Manchester University. He obtained a copy of the *Programmer’s Handbook* by Alan Turing and wrote a long letter to Turing on his plans:

... It would be a great convenience to say the least if the notation chosen were intelligible as mathematics when printed by the output ... once the suitable notation is decided, all that would be necessary would be to type more or less ordinary mathematics and a special routine called, say, ‘Programme’ would convert this into the necessary instructions to make the machine carry out the operations indicated. This may sound rather Utopian, but I think it, or something like it, should be possible ...

These ideas were very much in line with those of Turing, who ensured that Strachey was offered an attractive job to make use of his programming skills which tempted him away from teaching. Later Strachey went on to set up and lead the Programming Research Group as part of the Computing Laboratory in Oxford until his untimely death in the 1970s.

Programs specifically designed to read, manipulate and output mathematical algebraic formulae in a manner similar to that envisaged by Strachey are now readily available. Perhaps the market leader is *Mathematica*, developed by Wolfram Research Inc., founded by Stephen Wolfram (who was at school and briefly a student in Oxford during the 1960s and 70s). This is enthusiastically used by researchers such as Dana Scott—previously at Oxford and now at Carnegie-Mellon University in the US—to help speed up mathematical investigations in both teaching and research. As well as symbolic representations of algebraic formulae, graphical output is also possible. The speed and capacity of personal computers and workstations has now enabled such programs to be accessible to many people even though the implementation consists of over 1 million lines of C code.

## 6 Recent Developments in the Algebra of Programs

Until relatively recently, rules of algebra followed from assumptions about the unknown quantities, assuming that the symbols represented numerical values for example. By the 20th century, this had been replaced by a more abstract view. The rules could be interpreted in a concrete context if required, but this is not necessary or even appropriate in all circumstances. Other mathematical objects may also be manipulated algebraically and computer programs themselves may be considered as such objects, if assigned a suitable mathematical semantics.

General algebraic laws of programming, applicable to many imperative programming languages, or subsets of them, have been proposed by researchers led by Prof. Tony Hoare, who took over the leadership of the Programming Research Group at Oxford in 1977 [6]. Many special purpose *process algebras* have been proposed, with sets of laws for their manipulation to enable formal reasoning about them. Perhaps the most widely used of these are Robin Milner's CCS and Hoare's CSP (Communicating Sequential Processes), both useful for reasoning about concurrent systems. The latter has inspired the development of the parallel programming language Occam. The laws for this language, developed by Bill Roscoe, Hoare and others [10], have been used to successfully verify the floating-point unit for the Inmos T800 Transputer, which has resulted in commercial success for Inmos Ltd. and prestige in the form of a Queen's Award for Technological Achievement.

More recent work has investigated the use of algebraic laws in the verification of compiler specifications [4], either compiling directly into object code for a machine defined by an interpreter in the high-level language being compiled, or via a *normal form* consisting of a very restricted subset of the high-level language that is close to the final implementation. The proofs may be mechanised by systems such as OBJ, designed by Prof. Joseph Goguen, now at Oxford, who undertook much of the theoretical underpinning in the area of initial algebras and order sorted algebra. This tool facilitates the algebraic manipulation of mathematical structures (such as programs), with semi-automation of some of the more

mundane aspects including associative and commutative (“AC”) matching.

Modelling is an important aspect of computer science. However many algebraic properties may be derived from models. To quote Hoare [5]:

A model of a computational paradigm starts with choice of a carrier set of potential direct or indirect observations that can be made of a computational process. A particular process is modelled as the subset of observations to which it can give rise. Process composition is modelled by relating observations of a composite process to those of its components. Indirect observations play an essential role in such compositions. Algebraic properties of the composition operators are derived with the aid of the simple theory of sets and relations.

The viability of such approaches has yet to be seen in widespread industrial practice. As Hoare says [5]:

The construction of a single mathematical model obeying an elegant set of algebraic laws is a significant intellectual achievement; so is the formulation of a set of algebraic laws characterising an interesting and useful set of models.

But neither of these achievements is enough. We need to build up a large collection of models and algebras, covering a wide range of computational paradigms, appropriate for implementation either in hardware or in software, either of the present day or of some possible future.

However the prospects for the use of algebraic techniques in the design and verification of computer based systems are promising if the techniques can become as familiar to engineers as school algebra is to many today.

## References

- [1] J. Gattégno. *Lewis Carroll: Fragments of a Looking Glass*. Thomas Y. Crowell Company, New York, 1976. Translated by R. Sheed.
- [2] R.T. Gunther. *Chemistry, Mathematics, Physics and Surveying*, volume I of *Early Science in Oxford*. Oxford, 1923. Printed for the Oxford Historical Society at the Clarendon Press.
- [3] R.T. Gunther. *Oxford Colleges and their Men of Science*, volume XI of *Early Science in Oxford*. Oxford, 1937. Printed for the author.
- [4] C.A.R. Hoare. Refinement algebra proves correctness of compiling specifications. In C.C. Morgan and J.C.P. Woodcock, editors, *3rd Refinement Workshop*, Workshops in Computing, pages 33–48. Springer-Verlag, 1991.
- [5] C.A.R. Hoare. Algebra and models. *ACM Software Engineering Notes*, 18(5):1–8, December 1993.

- [6] C.A.R. Hoare et al. Laws of programming. *Communications of the ACM*, 30(8):672–687, August 1987.
- [7] A. Hodges. *Alan Turing: The Enigma*. Simon & Schuster, New York, 1983.
- [8] R.A. Hyman. *Charles Babbage: Pioneer of the Computer*. Oxford University Press, 1982.
- [9] N.K. Nichols. Obituary: Professor Leslie Fox, C. Math., FIMA. *IMA Bulletin*, 29:175–176, November/December 1993.
- [10] A.W. Roscoe and C.A.R. Hoare. Laws of Occam programming. *Theoretical Computer Science*, 60:177–229, 1988.