



# Histories of algorithms: Past, present and future

Maarten Bullynck

► **To cite this version:**

Maarten Bullynck. Histories of algorithms: Past, present and future. *Historia Mathematica*, Elsevier, 2015, 43 (3), pp.332 - 341. 10.1016/j.hm.2015.07.002 . halshs-01215943

**HAL Id: halshs-01215943**

**<https://halshs.archives-ouvertes.fr/halshs-01215943>**

Submitted on 15 Oct 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Histories of algorithms: Past, present and future

- [Maarten Bullynck](#)
- Université Paris 8, France

The book under review, first published some twenty years ago, is the product of a group of French historians of mathematics: Jean-Luc Chabert, Evelyn Barbin, Michel Guillemot, Anne Michel-Pajus, Jacques Borowczyk, Ahmed Djebbar et Jean-Claude Martzloff. An English translation by Chris Weeks was published by Springer in 1999 ([Chabert et al., 1999](#)) and a second French edition is now available and will be reviewed here. It was at the time of its first appearance—and is still—the sole book-length study of the history of mathematical algorithms. Therefore this second corrected and extended edition of the French original is most welcome.

The original edition of *Histoire d'algorithmes* was published at a time of developing interest in the history of algorithms within the mathematics community and it was, in part, conceived for use in the mathematics classroom. The historical range of the book is impressive; from Ancient Mesopotamia and Egypt to modern computer developments, this volume endeavours to forge a history of mathematics through highlighting the role of algorithms. The book contains 15 chapters and closes with short biographies of the authors discussed and a general index. Each chapter opens with a general overview that introduces the reader to the chapter's topic. The topic is illustrated and developed through a series of excerpts taken (or translated into French) from the original, historical documents. The text is always followed by a rewriting in terms of elementary modern mathematics and by some analysis and *posthistoire* of the algorithm. While the main text of the book has been largely kept as it was in the first edition, this more recent *posthistoire* has been updated for this second edition.

The book, one finds, falls naturally into two parts. The first (chapters 1–5), comparative in nature, focuses on the structure of an algorithm at different times in different cultures and traditions, principally, though not exclusively, pre-modern and non-Western. The second part remains largely within the realm of modern Western mathematics (17th–20th centuries) and presents a more or less linear story of how certain kinds of algorithms used in numerical analysis were developed and generalized.

The comparative first five chapters cover arithmetical operations (Ch. 1), magic squares (Ch. 2), false position (Ch. 3), Euclid's algorithm (Ch. 4) and the calculation of  $\pi$  (Ch. 5). Topics in the second numerical analysis part of the book include Newton's method (Ch. 6), approximate numerical solutions to equations (Ch. 7), systems of linear equations (Ch. 9), interpolation (Ch. 10), quadratures (Ch. 11), differential equations (Ch. 12), approximation of functions (Ch. 13), and acceleration of convergence (Ch. 14). Interspersed with these topics from numerical analysis are a chapter on algorithms in number theory (Ch. 8) and a chapter on the development of a mathematical definition of algorithm in the nineteen-thirties by Gödel, Post, Church, Turing and Kleene (Ch. 15).

Finally, this second edition features a new final chapter, “Epilogue. Ecriture, temps, hasard”. This “epilogue” discusses some recent topics not covered in the first edition of the book, such as programming languages, time complexity, probabilistic algorithms, pseudo-random generators and quantum algorithms. The choice of topics in this last chapter reflects the interests of the authors in recent developments in computer science. The treatment is quite summary and the chapter as a whole is rather an afterthought, lacking the coherence of the rest of the book. It does not feature

excerpts of original papers, except for Edouard Lucas' description of the Tower of Hanoi problem. In particular, it would have profited from some of the literature on the history of computing, such as [Knuth and Pardo \(1977\)](#) and [Wexelblatt \(1981\)](#) on programming languages or [Mahoney \(1997\)](#) on theories of computation, not to mention more recent work.

In this second French edition, many errors found during the translation of the English edition or pointed out by reviewers have been corrected. Nevertheless, the book still contains quite a number of inaccuracies that more careful editing might have prevented. These are mostly due the use of secondary sources that have not been double-checked. For example, the Liber Abaci was first published in 1202 not in 1228 (p. 290); the ENIAC was not an analogue (!) machine, but rather one of the first digital computers (p. 436); Lambert's article dates from 1768, not 1761 (p. 188); BNF is the abbreviation of Backus Normal Form, only later did the acronym come to mean Backus Naur Form (p. 539), etc.

The bibliography remains rather patchy and is inconsistent in style throughout the chapters (details of reprints are not given, page numbers are often missing, first names of author are not always given, etc.). In this second edition a restricted number of more recent bibliographic references have been added, though no effort has been made to systematically update the bibliography. Besides the references to primary technical sources the book only rarely references the secondary and historical literature. Also, in some cases, the authors use older editions and translations of ancient texts (such as Colebrooke's Bhaskara translation or Peyrard's Euclid edition), while better, modern editions are available.

Nonetheless, despite these problems, this is an excellent source-book, rich in texts and themes, eminently suitable for use in the classroom. The reproduction or translation of the original texts greatly helps an appreciation of the historical texture and dimension of algorithms and may in many cases be an excellent point of departure for adding a historical dimension to mathematical lectures.

And yet there remains a sense of frustration with this book today, in that it no longer reflects the state of the art in the historiography of algorithms. Adapting some of the historical concepts, tools and other innovations of the last twenty years would have added considerably more depth and improved the treatment of some topics in this second edition of *Histoire d'algorithmes*. Though it is understandable that the book was not entirely reworked for this new edition, it is unfortunate that the occasion to engage with more recent work in a foreword or afterword has been missed. We would like to make up for this omission by giving an overview of what has been happening in the history of algorithms in the last 20 years and point to some work still in progress.

**The history of algorithms in computer science.** At its initial appearance in 1994, *Histoire d'algorithmes* was part of a rekindled interest in algorithms within the historiography of mathematics, although the origin of the study of algorithms as a mathematical and historical object reaches back to the decades between 1960 and 1980 when computer science was trying to establish itself as an independent academic discipline. <sup>1</sup> It is in this process of disciplinarisation that we find the first serious historical reflections on algorithms and other basic concepts of computer science. In this respect it is informative to take a look at two important contributions to the history of algorithms that were written by computer scientists in the early 1970s. Both contributions still serve, albeit in different ways, as points of reference.

Donald E. Knuth was one of the people who had been promoting the concept of algorithm as a basic notion in computer science with his classic three-volume *The Art of Computer Programming* (1968–1973). During the writing of this he seized the occasion of a special twentieth-anniversary issue of the Communications of the Association for Computing Machinery to publish his pioneering paper “Ancient Babylonian Algorithms” ([Knuth, 1972](#)). The opening phrase of the article explicitly states his motivation: “One of the ways to help make computer science respectable is to show that it is deeply rooted in history, not just a short-lived phenomenon” (p. 671). Drawing almost exclusively on Otto Neugebauer's work, Knuth analyses some Babylonian mathematical texts with the help of his modern understanding of computer arithmetic and programming. Knuth especially foregrounds

the fact that a certain “algebraic” (we would rather say, procedural) format existed for expressing algorithms, that they used floating point arithmetic and that memory operations and even a kind of assignment are clearly distinguished. Knuth exclaims: “The above procedure reads surprisingly like a program for a stack machine like the Burroughs B5500” (p. 674), where one needs to know that the B5500 machine was a quite advanced computer for its time that had been designed with a high-level programming language in mind. As [Knuth \(1970\)](#) had shown two years before in an analysis of John von Neumann's first computer program (dated 1946), the development of a formal device for expressing algorithms and memory management during a computation is highly non-trivial. <sup>2</sup>

Sharing in the disciplinarian vogue for computer science, that same year Herman H. Goldstine published his book *The Computer from Pascal to von Neumann* ([Goldstine, 1972](#)). Goldstine had trained as a mathematician, then had specialised in ballistics and become an officer in the Ballistics Research Laboratory at the Aberdeen Proving Ground during World War II. In that function he was the liaison officer with the team at the Moore School of Electrical Engineering, University of Pennsylvania, that was building the ENIAC, one of the world's first computers. Goldstine was responsible for recruiting John von Neumann for the group and, after the war, continued his collaboration with von Neumann, co-authoring various important founding documents in computing. Reflecting his own experience, Goldstine's book prominently featured applied mathematics (Part II, chapter 4: “Numerical Analysis”). This contains the bold—and historically inaccurate—statement that “the state of numerical mathematics stayed pretty much the same as Gauss left it until World War II” (p. 287).

Five years later, Goldstine published a book-length elaboration of that statement, *A History of Numerical Analysis* ([Goldstine, 1977](#)). In the introduction, Goldstine too thanks Otto Neugebauer for “help and inspiration in the preparation of this book”. Describing theories and algorithms now belonging to the toolkit of a numerical analyst, Goldstine's story opens with Napier's logarithms and goes on in a linear progression until Gauss (and a final chapter quickly runs over Jacobi, Cauchy and Hermite). Bringing in the great names from the history of mathematics, Goldstine constructs an illustrious pedigree for the new discipline of numerical analysis, without even mentioning the more practical and down-to-earth problems of manual calculation or the effects of the birth of the digital electronic computer. For many (including the authors of the book under review) *A History of Numerical Analysis* remains the classic source-book on the history of algorithms used in numerical analysis.

Both Knuth's Babylonian paper and Goldstine's books attempted to connect the history of the young discipline with an older tradition in the history of mathematics, the ‘great tradition’ of Göttingen for which Neugebauer stood. Where Goldstine's book, adapting itself to a mathematical mold, can be regarded as the (pre)history of a mathematical discipline, viz. numerical analysis, Knuth's paper, in contrast, brought the tools and concepts of computer science into a classic part of the history of mathematics.

In the 1980s and 1990s, around the time that computer science departments were being established at many universities and the notion of “algorithm” was slowly becoming a topic within the secondary-school curriculum, some professional historians of mathematics started addressing algorithms as well. To some extent independently of the developments in computer science, these historians had also discovered the algorithm as an historical mathematical object in its own right.

**The history of algorithms in ancient cultures.** The concept of “algorithm” became an analytical tool of historiographic research and innovation because it allowed to connect with many other tendencies in the renewal of the historiography of mathematics since the 1990ies. Leaving a perspective centered on the Western World and on “modern” mathematics that privileged demonstration, abstraction and innovation, historians of mathematics have started looking at the practices, contexts, traditions and cultures of mathematics and have tried to consider mathematical practices as one among many structured rational practices.

Focussing on algorithms has proven very fruitful in the historiography of Ancient non-Western

mathematics. Among the first to draw attention to ways of algorithmic thinking are Jim Ritter in his work on Egyptian and Babylonian mathematics and Karine Chemla in her work on ancient Chinese mathematics.

In order to bring out both the differences and similarities between structures found in Egyptian and Babylonian procedural texts, [Ritter, 1989](#) and [Ritter, 2004](#) uses a simple symbolic rewriting system (and sometimes a schematic flowchart) to construct a context that may “have an historical basis while others serve a heuristic purpose for the modern student” ([Ritter, 2004](#), p. 178). This formal analysis helps to follow the gradual complexification of examples and the re-use of already known “subroutines”, it also maps the operations of writing down and recalling intermediate results. It is an analytical tool that brings out algorithmic structures in ancient texts and may be used to compare them. Another aspect should be insisted upon, such an analysis is not constricted to mathematical texts, but can be applied to other formally tailored texts. Ritter's work brings out the formal similarity of rational structures occurring in a variety of cuneiform texts, not only mathematical, but also medical or juridical texts. This algorithmic or procedural style is thus transversal to (modern-day) disciplinary boundaries.

The focus of Chemla's work has been the structure of procedural mathematical texts as they codify and build ever more general or abstract arguments (e.g. [Chemla, 1987a](#), [Chemla, 1987b](#), [Chemla, 1991](#), [Chemla, 1992](#) and [Chemla and Shushun, 2004](#)). In many mathematical traditions the dynamic interaction between problems and procedures is a central part of mathematical activity. On the one hand, a procedure is slowly explained and generalised through its application in a number of problems. On the other hand, a sequence of problems gradually mobilises combinations of subprocedures explained before, and a given problem may even be solved by different combinations of subprocedures. Procedures thus acquire a certain generality. For instance in Liu Hiu's comments on the classic *Nine Chapters on mathematical procedures* this generality is stressed throughout by the introduction of an “abstract” vocabulary that brings out the ties that exist between the several chapters. Thus, the comments introduce a level of generality and ways of verification that, although they do not equate with the abstraction of classic demonstration, are fundamental devices of mathematics too.

Ritter's and Chemla's work have led to other studies in the fields of Egyptian, Chinese, Indian and Babylonian mathematics. See, e.g., the work of A. [Imhausen, 2002](#) and [Imhausen, 2003](#), A. [Bréard, 2000](#) and [Bréard, 2008](#), A. [Keller \(2006\)](#), [C. Proust \(2007\)](#).<sup>3</sup> There has been a warning that reading ancient mathematics with algorithmic lenses risks of running into the same anachronistic misreadings as reading it with algebraic glasses ([Høyrup, 2008](#)). Most researchers, however, have avoided that pitfall because they used a procedural reading as an analytical, nearly linguistic tool that helps to foreground particular aspects of old texts, not as a goal per se.

This work has also fruitfully challenged and innovated paradigms governing the history of ancient Greek mathematics. In ([Ritter and Vitrac, 1998](#)) two styles in Greek mathematics were discerned: a demonstrative (e.g. Euclid) and an algorithmic one (e.g. in Heron and Diophantus). While the first style has been classically given higher status than the second one by historians, the algorithmic tradition should also be regarded as a constituent part of Greek mathematics. Looking at Greek mathematics from this perspective brings it in closer contact with “oriental” traditions in mathematics such as the Egyptian or Babylonian styles of mathematics that display a more algorithmic character.<sup>4</sup> Building on the idea of procedural texts and their development through examples, [Bernard and Christianidis \(2011\)](#) have shown how the composition of Diophantus' *Arithmetic* progresses example by example from simple to complex with an accumulative introduction of new devices of invention, mirroring the rhetorical construction of a text as used in classic Greek education.

As the quoted studies abundantly show, an important characteristic of the algorithmic style is the use of sequences of (numerically computed) problems and examples to explain and gradually develop an algorithm. Example or problem series are in fact a recurrent feature in nearly all

mathematical traditions and they properly belong to the main body of mathematical knowledge.<sup>5</sup> In particular, these series have a long half-life, they are often copied, recycled and recombined<sup>6</sup> and their transmission and “recycling” is an important topic of modern historiography. Only in late 18th century Europe was a (pedagogic) separation enforced between the textbook and theory on the one hand and the series of problems and examples on the other hand, making the latter seem like a minor matter.

These sequences or series are of course part of an educational context in which a procedure is learned through repeated exercise, but where a procedure also becomes more flexible in the hands of a pupil who proceeds through more difficult, intricate or compounded problems. It is also an expression of the master's knowledge and skill at developing series of problems that go through all useful variations and complexities, demonstrating the possibilities of combination and modification, while hinting at links with other topics. Such series of problems can show the various forms of a procedure, the possible extensions of a procedure, its combinations with other procedures, etc. It can help to broaden or generalise procedures and the concepts involved, or it can specialise to intricate particular cases where a number of procedures have to be carefully fitted and tied together.

**The history of algorithms in 20th century mathematics.** Contrary to Goldstine's often reproduced account in *A History of Numerical Analysis*, the first formations of numerical analysis as a discipline come long before the end of World War II, John von Neumann's contributions or the birth of the digital computer. Goldstine's narrow focus on the 16th to 18th century and on the greater names in mathematics has been criticised by [Folta \(1988\)](#), and, more recently, by [Tournès \(2014\)](#). Both pointed out that computational practices as now codified in numerical analysis may be found throughout the centuries, but very often not in “pure” mathematics, rather in fields of practical, mixed or applied mathematics. A history of numerical analysis has to take this aspect into account. As a discipline, Tournès argues, numerical analysis was instead to emerge in the late 19th century and to have a further momentum of disciplinarisation after World War I.<sup>7</sup> Indeed, the first World War and its computational needs, especially in ballistics, seems to have stimulated a first systematic review of the mathematical literature in order to find efficient methods of computation ( [Gluchoff, 2011](#) and [Aubin and Goldstein, 2014](#)). This led to the publication of the first manuals on practical calculation during the Interwar period.

Newer historiography has tried to integrate the immediate practical context in which many of the computational techniques were first developed into the mathematical narrative. In the modern period, there are often forms of ‘mathematical workplaces’ involved, to borrow an expression from ([Epple, 2000](#)). In these workplaces a group of persons, who may or may not be trained mathematicians, take up, apply and use specific parts of mathematical knowledge in a technological environment. The “technological” environment may vary from crafts or schools that practise account keeping or surveying to highly organised industrial or military complexes of research and development. There exist a number of interesting studies on the mathematical practitioners and human computers in such institutionally and socially defined contexts as the observatory ([Wepster, 2010](#) and [Aubin et al., 2010](#)), the computation bureau ([Croarken, 1990](#) and [Grier, 2005](#)), K. Pearson's Biostatistics Lab (J. Barrow-Green in [Aubin and Goldstein, 2014](#)) or on German computational work in aerodynamics ([Epple et al., 2005](#)).

Many traditions of computation are intrinsically intertwined with the use and production of tables and/or instruments. Tables or instruments are part of the materiality of calculation, as such, they shape the way mathematical procedures are thought up and transformed. The material aspect is often hard to reconstruct because it frequently relies on an implicit, tacit know-how that is not codified into textbook or manual but rather passed along orally or by experience. Although the book under review refers to this materiality in its subtitle, “from the pebblestone to the computer chip” (“du caillou à la puce”), it is generally absent in the book's treatment of algorithms. The pivotal role of tables in performing Babylonian and Egyptian arithmetics is only mentioned in passing (pp. 13–25), the discussion of interpolation methods leaves little space for the production and use of large



(trigonometric, logarithmic etc.) tables is (p. 355f.), and the digital computer hardly appears at all. Recent studies, however, have focussed on the materiality of and the tools used in computing and mathematics.

In 2003 an edited volume ([Campbell-Kelly et al., 2003](#)) gave a first introduction to the histories of mathematical tables. Among the topics treated were the industrial division of labour in de Prony's or Babbage's table projects, the Committees of mathematical tables around 1900 and modern day spreadsheets. The book rather introduced the question and hardly exhausted it, besides its focus was mainly on the English-speaking world in the modern period. A more ambitious approach was formulated in a project funded by the French ANR and directed by D. Tournès who had already been working on the development of tractional instruments and nomographs in 19th and 20th century mathematics ([Tournès, 2000](#) and [Tournès, 2009](#)). The project *History of Numerical Tables* (HTN) wants to investigate the role of numerical tables in various traditions of mathematics. Since 2008 over twenty researchers from various countries have regularly met to prepare a book on numerical tables. The cross-cultural and cross-disciplinary discussions on tables have opened up a number of very general, methodological questions on the structure, production, use and transmission of tables. <sup>8</sup> The intertwining of mathematical procedures and numerical tables that is so typical of Babylonian mathematical culture, is also found in some schools of medieval Arabic science and in some early modern European mathematicians. Further, some domains of both applied and pure mathematics are characterised by a strong interaction between procedures of computation and numerical tables, e.g., astronomy, ballistics, 18th century elliptic functions or number theory. An overview of the project can be found in [Tournès \(2011\)](#).

Much of the interest in algorithms is inspired directly or indirectly by the advent of the digital electronic computer, but surprisingly few historical studies exist on the particular interaction between algorithms, the computer and its users or programmers. It has often been argued that mathematics has or is changing since the advent of digital computing, but those historical studies have mostly focussed on the history of modelling and simulation (e.g. [Galison, 1997](#) or [Dahan-Dalmedico, 2001](#)), or on the mechanisation of proof (e.g. [McKenzie, 1999](#)). Concrete studies on how the computer has impacted on mathematics or on the use of algorithms are still scarce. It is still mostly a history written by its actors who write a short, often personal, mostly linear story of the development of an algorithm in a particular context.

Admittedly, the material context of the computer is often difficult to take into account in a reasonable and apprehensive way. Some of this know-how has not been explicitly recorded and much of it may quickly lead to over-detailed and technical stories. However, the co-evolution of algorithms and the computer is an intriguing field that invites historiographic exploration. For one of the first electronic digital computers, the ENIAC (1946), a number of such studies have appeared. [Bullynck and De Mol \(2010\)](#) and [De Mol, Carlé and Bullynck \(2015\)](#) have shown the reciprocal dependencies that exist between the machine, its users and the mathematics used and transformed. It analyses questions as how speed and accuracy of an algorithm have to be balanced against simplicity of control and program expression, or how the architecture of a computer conditions the short and long term memory management of a procedure. A key idea is that the early computers functioned as a kind of meeting ground where people from various backgrounds came together and interacted and where specific styles of computing could develop. [Haigh et al. \(2014\)](#) bring a reconstruction of the production process and life cycle of the first Monte Carlo program that ran on the ENIAC rewired into stored-program mode.

Such studies can cover a longer period too, and track the influence of the rapid evolution of computer hardware and of programming languages and techniques on the choice, sequences and variations of algorithms within a specific discipline. Franck [Varenne, 2007](#) and [Varenne, 2008](#) has shown, for instance, that increasing speed and memory of computers, combined with the development of object-oriented programming, made it possible to study not only of the structure of plants, but also of the temporal unfolding of the structure itself through computer modelling. In a

field that has gained in prominence due to cryptographic applications, computational number theory, [Bullynck \(2009\)](#) has described how a complex taxonomy of various factorisation algorithms slowly developed, not only echoing the size of the number to be factorised but also the variety of computer devices, from parallel to serial architecture, from hand-held calculators to supercomputers. In yet another field, that of chess programs, [Ensmenger \(2012\)](#) showed that over the years a rather brute-force minimax algorithm grew to be preferred over subtler, heuristic algorithms.

**Conclusion.** It may be clear by now that the concept of algorithm has borne much fruit in recent historiography, but also that much work remains to be done. The book under review, *Histoire d'algorithmes*, still is a nice textbook to introduce the student to the fascinating field of the history of algorithms. This review essay has tried to supplement it by introducing a number of topics that have been explored in the recent historiography of algorithms and serves as an invitation to the reader to discover this thriving field. For those who want to dig deeper into a specific topic, the rather extensive list of references listed here may provide orientation and inspiration for further reading.

## Acknowledgements

This review essay was begun in collaboration with Jim Ritter while we were teaching a History of Algorithms course at Université Paris 8. Although I alone now feature as the author, this review owes much to this collaboration and to the inspiring discussions I had with Jim. All errors, interpretations and misconceptions are, of course, solely my responsibility.

## References

- Aubin, D., Bigg, C., Sibum, O. (Eds.), 2010. *The Heavens on Earth: Observatories and Astronomy in Nineteenth-Century Science and Culture*. Duke University Press, London.
- Aubin, D., Goldstein, C. (Eds.), 2014. *The War of Guns and Mathematics: Mathematical Practices and Communities through World War I in France and its Western Allies*. Springer, Berlin, New York.
- Bernard, A., Christianidis, J., 2011. A new analytical framework for the understanding of Diophantus's *Arithmetica* I–III. *Arch. Hist. Exact Sci.* 66 (1), 1–69.
- Bréard, A., 2000. Les problèmes comme champ d'interprétation des algorithmes dans les Neufs Chapitres sur les procédures mathématiques et leurs commentaires. *Oriens-Occidens* 3, 189–234.
- Bréard, A., 2008. A summation algorithm from 11th century China. Possible relations between structure and argument. In: Beckmann, A., Dimitracopoulos, C., Löwe, B. (Eds.), *Logic and Theory of Algorithms. Proceedings of the 4th Conference on Computability in Europe*. In: *Lecture Notes in Computer Science*, vol. 5028. Springer, Berlin.
- Brezinski, C., Wuytack, L., 2001. *Numerical Analysis: Historical Developments in the 20th Century*. Elsevier, Amsterdam.
- Bullynck, M., 2009. Reading Gauss in the computer age: on the U.S. reception of Gauss's number theoretical work (1938–1989). *Arch. Hist. Exact Sci.* 63 (5), 553–580.
- Bullynck, M., De Mol, L., 2010. Setting-up early computer programs. D.H. Lehmer's ENIAC computation. *Arch.Math. Log.* 49 (2), 123–146.
- Campbell-Kelly, M., Robson, E., Croarken, M., Flood, R., 2003. *The History of Mathematical Tables: From Sumer to Spreadsheets*. Oxford University Press, Oxford. See also D. Tournès' review in *Nuncius. Journal of the History of Science* 20 (1)



(2005) 275–276.

- Chabert, et al., 1999. *A History of Algorithms. From the Pebble to the Microchip*. Springer, Berlin.
- Chemla, K., 1987a. L'aspect algorithmique récurrent dans les mathématiques. In: Dhombres, J. (Ed.), *Cahiers d'Histoire et de Philosophie des Sciences*, vol. 20, pp. 86–104.
- Chemla, K., 1987b. Should they read FORTRAN as if it were English? *Bull. Chinese Stud.* 1 (2), 301–316.
- Chemla, K., 1991. Theoretical aspects of the Chinese algorithmic tradition (first to third centuries). *Hist. Sci.* 42, 75–98.
- Chemla, K., 1992. Résonances entre démonstration et procédure. Remarques sur le commentaire de Liu Hui (3e siècle) aux Neufs Chapitres sur les Procédures Mathématiques (1er siècle). *Extrême-Orient, Extrême-Occident* 14, 91–129.
- Chemla, K., Shushun, G., 2004. *Les Neuf chapitres. Le Classique mathématique de la Chine ancienne et ses commentaires*. Dunod, Paris.
- Croarken, M., 1990. *Early Scientific Computing in Britain*. OUP, Oxford.
- Dahan-Dalmedico, A., 2001. History and epistemology of models: meteorology as a case-study (1946–1963). *Arch. Hist. Exact Sci.* 55, 395–422.
- De Mol, L., Carlé, M., Bullynck, M., 2015. Haskell before Haskell. An alternative lesson in practical logics of the ENIAC. *J. Log. Comput.* 25 (4), 1011–1046.
- Epple, M., 2000. Genies, Ideen, Institutionen, mathematische Werkstätten: Formen der Mathematikgeschichte. Ein metahistorischer Essay. *Math. Semesterber.* 47, 131–163.
- Epple, M., Karachalios, A., Remmert, V.R., 2005. Aerodynamics and mathematics in national socialist Germany and fascist Italy: a comparison of research institutes. *Osiris* 20 (1), 131–158.
- Ensmenger, N., 2012. Is Chess the Drosophila of AI? A social history of an algorithm. *Soc. Stud. Sci.* 42 (1), 5–30.
- Folta, J., 1988. Some remarks on the history of numerical analysis especially in the area of Prague. *Llull* 11, 217–233.
- Galison, P., 1997. *Image and Logic: A Material Culture of Microphysics*. Chicago University Press, Chicago.
- Gluchoff, A., 2011. Artillerymen and mathematicians: Forest Ray Moulton and changes in American exterior ballistics, 1885–1934. *Hist. Math.* 38 (4), 506–547.
- Goldstine, H.H., 1972. *The Computer. From Pascal to von Neumann*. Princeton University Press, Princeton.
- Goldstine, H.H., 1977. *A History of Numerical Analysis. From the 16th Through the 19th Century*. Springer, New York, Heidelberg.
- Grier, D.A., 2005. *When Computers Were Human*. Princeton University Press, Princeton.
- Gupta, G., 2007. Computer science curriculum developments in the 1960's. *IEEE Ann. Hist. Comput.* 29 (2), 40–54.
- Haigh, Th., Priestley, M., Rope, C., 2014. Los Alamos bets on ENIAC: nuclear Monte Carlo simulations 1947–8. *IEEE Ann. Hist. Comput.* 36 (3), 42–63.
- Heeffers, A., 2006. Van precepten tot vergelijkingen: de conceptuele ontwikkeling van de symbolische algebra in de zestiende eeuw. PhD thesis. Universiteit Gent.
- Høyrup, J., 2002. *Lengths, Widths, Surfaces: A Portrait of Old Babylonian Algebra and Its Kin*. Springer, Berlin.
- Høyrup, J., 2008. Algorithm concept – tool for historiographic interpretation or Red Herring? In: Beckmann, A., Dimitracopoulos, C., Löwe, B. (Eds.), *Logic and Theory*

- of Algorithms. Proceedings of the 4th Conference on Computability in Europe. In: Lecture Notes in Computer Science, vol. 5028. Springer, Berlin, pp. 261–272.
- Hudeček, J., 2012. Ancient Chinese mathematics in action: Wu Wen-Tsun's nationalist historicism after the Cultural Revolution. *East Asian Sci. Technol. Society* 6 (1), 41–64.
  - Imhausen, A., 2002. The algorithmic structure of the Egyptian mathematical problem texts. In: Steele, J., Imhausen, A. (Eds.), *Under One Sky: Astronomy and Mathematics in the Ancient Near East*. In: *Alter Orient und Altes Testament*, vol. 202. Ugarit-Verlag, Münster, pp. 147–161.
  - Imhausen, A., 2003. *Ägyptische Algorithmen. Eine Untersuchung zu den mittelägyptischen mathematischen Aufgabentexten*. Harassowitz, Wiesbaden.
  - Keller, A., 2006. *Expounding the Mathematical Seed: A Translation of Bhaskara I on the Mathematical Chapter of the Aryabhatiya*. Birkhäuser, Basel.
  - Knuth, D.E., 1970. Von Neumann's first computer program. *Computer Surv.* 2 (4), 247–260.
  - Knuth, Donald, 1972. Ancient Babylonian algorithms. *Commun. ACM* 15 (7), 671–677.
  - Knuth, D., Pardo, L.T., 1977. The early development of programming languages. In: Belzer, J., Holzman, A.G., Kente, A. (Eds.), *Encyclopedia of Computer Science and Technology*, vol. 7. Marcel Dekker, pp. 419–493.
  - Mahoney, M., 1997. Computer science: the search for a mathematical theory. In: Krige, John, Pestre, Dominique (Eds.), *Science in the 20th Century*. Harwood, Amsterdam. Chapter 31; Reprinted in: Mahoney, M.S., *Histories of Computing*, edited by Th. Haigh. Harvard University Press, Cambridge, MA, 2011, pp. 128–146.
  - McKenzie, D., 1999. Slaying the Kraken: the sociohistory of a mathematical proof. *Soc. Stud. Sci.* 29 (1), 7–60.
  - Proust, C., 2007. *Tablettes mathématiques de Nippur. I: Reconstitution du cursus scolaire. II: Edition des tablettes conservées à Istanbul. Translittération des textes lexicaux et littéraires par Antoine Cavigneaux. Préface de Christian Houzel. Varia Anatolica*, vol. 18. IFEA – De Boccard, Istanbul.
  - Ritter, J., 1989. *Babylone – 1800; Chacun sa vérité: les mathématiques en Égypte et en Mésopotamie*. In: Serres, M. (Ed.), *Éléments d'Histoire des Sciences*. Bordas, Paris, pp. 16–37, pp. 38–61.
  - Ritter, J., 2004. Reading Strasbourg 368: a thrice-told tale. In: Chemla, K. (Ed.), *History of Science, History of Text*. Springer, Dordrecht, pp. 177–200.
  - Ritter, J., Vitrac, B., 1998. *Pensée grecque et Pensée "orientale"*. In: Mattéi, J.F. (Ed), *Encyclopédie Philosophique*, vol. IV, pp. 1233–1250.
  - Tournès, D., 2000. Pour une histoire du calcul graphique. *Rev. Hist. Math.* 6 (1), 123–157.
  - Tournès, D., 2009. *La construction tractionnelle des équations différentielles*. Blanchard, Paris.
  - Tournès, D. (Ed.), 2011. *History of Numerical and Graphical Tables*, organized by Renate Tobies and Dominique
  - Tournès. *Oberwolfach Reports*, vol. 8, pp. 639–689.
  - Tournès, D., 2014. Mathematics of engineers: elements for a new history of numerical analysis. In: *Proceedings of the International Congress of Mathematicians*, vol. 4. Seoul, August 13–21, pp. 1255–1273.
  - Varenne, F., 2007. *Du modèle à la simulation informatique*. Vrin, Paris.
  - Varenne, F., 2008. *Modèles et simulations: pluriformaliser, simuler, rémathématiser*. *Matière première* 1 (3), 153–180.
  - Wepster, S., 2010. *Between Theory and Observations: Tobias Mayer's Explorations*

of Lunar Motion, 1751–1755.

- Springer, Berlin.
- Wexelblatt, R.L. (Ed.), 1981. History of Programming Languages. Academic Press, New York.
- Wu, W., 1980. Shuxue de jixiehua (Mechanization of Mathematics). Baike zhishi (Encyclopedic Knowledge) 3, 41–44.

1

See e.g. [Gupta \(2007\)](#). The construction of computer science as a discipline is still very much a live topic in the historiography of computing, see the recent panel at the 2013 SIGCIS Workshop organized by Janet Abbate, <http://www.sigcis.org/node/381>.

2

Independently of Knuth, the Chinese topologist Wu Wenjun, working in a computer factory during the Cultural Revolution, had become interested in Ancient Chinese mathematics and recognized its algorithmic nature. If Knuth in 1972 had aimed to legitimize algorithms by showing their venerable age, in 1980 Wu Wenjun conversely used his recognition of the algorithmic (“mechanized”) nature of Ancient Chinese mathematics to justify his own pioneering work in automatic proof procedures ([Wu \(1980, 43–44\)](#), cited in [Hudeček \(2012, 55\)](#)): “We set out the question and came up with a method of solution under inspiration from Ancient Chinese algebra. The reason is actually easy to understand; Chinese ancient mathematics was basically a mechanized mathematics.... The work of the present author on mechanization of mathematics is precisely a product of inspiration from these ideas and results, it is a direct continuation of our mathematics from the Nine Chapters up to the Song and Yuan periods.” (This footnote is due to Jim Ritter.)

3

It may be remarked that the treatment of false position in the book under review (pp. 99–102) could be updated, fine-tuned and often improved by consulting this recent work on algorithms in non-Western traditions, viz. ([Høyrup, 2002](#), pp. 59–80; 101–107) and ([Ritter, 2004](#), 179–184) for Babylonia; ([Imhausen, 2003](#), pp. 39–42) and ([Ritter, 2004](#), pp. 184–186) for Egypt; ([Chemla and Shushun, 2004](#), pp. 549–599) for China.

4

Since 2009, an ANR-funded project “Pratiques déductives algorithmiques dans les mathématiques pré-algébriques” (ALGO) directed by F. Acerbi and B. Vitrac has brought together researchers on ancient mathematics to address these questions more systematically. See <http://algo.hypotheses.org.gate3.inist.fr/a-propos>.

5

Under the direction of A. Bernard at HASTEC a research group has been focussing on the role of series of problems in different (algorithmic) traditions (“Les séries de problèmes, un genre au croisement des cultures”). See <http://problemata.hypotheses.org.gate3.inist.fr/>.

6

E.g., during his work on his PhD, A. [Heeffer \(2006\)](#) has drawn up a database of problems and their copies and modifications in early algebra textbooks to study their migration and

dissemination.

7

The list of early university courses on numerical analysis given in ([Brezinski and Wuytack, 2001](#), pp. 4–20) confirms this.

8

See <http://www.sphere.univ-paris-diderot.fr/spip.php?rubrique64>.