



HAL
open science

Programming men and machines. Changing organisation in the artillery computations at Aberdeen Proving Ground (1916-1946)

Maarten Bullynck

► **To cite this version:**

Maarten Bullynck. Programming men and machines. Changing organisation in the artillery computations at Aberdeen Proving Ground (1916-1946). 2015. halshs-01541616

HAL Id: halshs-01541616

<https://shs.hal.science/halshs-01541616>

Preprint submitted on 19 Jun 2017

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.

Programming men and machines. Changing organisation in the artillery computations at Aberdeen Proving Ground (1916-1946)

Maarten Bullynck (Paris 8)

The United States' decision to enter World War I marks the beginning of a new and close alliance between science and the military. On June 19, 1916 the National Research Council was organized to encourage “the employment of scientific methods in strengthening the national defense”.¹ This employment would come in many forms, ranging from technological developments to the application of statistical control procedures and psychological tests for managing the commodities needed during the war effort. One of the earliest employments of science was the reform of the mathematical procedures used in ballistics.

While at first sight this reform can be regarded as a purely mathematical one, the preference of one method over another, it also marks the beginning of a new organisation of men, tables and machines for computational labour. In the scientific articles that discuss the mathematical reform this aspect remains implicit, but the technical reports and field manuals speak more clearly of the effect of mathematical forms on the organization of manual calculation. This new organization of computational labour would imprint itself on the minds of the artillery officers. When, starting from the 1930s, new machinery, analogue or digital, will be deployed to aid in the artillery computations, these dispositives will still guide them in using and describing the new machines, having an impact on the hardware architecture and/or the utilisation and programming of the early computers.

1. New ways of organizing human computation labour after World War I

During the first World War the astronomer Forrest R. Moulton brought more advanced mathematical techniques to the computation of ballistic tables, replacing older techniques that had become obsolete in part due to the advances in weaponry. Moulton introduced a method called *numerical integration* that was refined during the years 1917-1928.² Another revolution, less explicit perhaps but with more consequences ultimately, went hand in hand with this mathematical reform: a reorganisation of the “assembly line” of ballistic computation, of the actual sequence of steps and actions during computation.

Moulton (1919, p. 50-51) started from the classic set of ballistic differential equations:

$$d^2x/dt^2 = -F dx/dt$$

$$d^2y/dt^2 = -F dy/dt - g$$

where $F = G(v) H(y) / C$ with $G(v = \sqrt{x'^2 + y'^2})$ the retardation function (air resistance, drag, etc.) for a standard projectile, $H(y)$ the altitude function

1 NRC Annual report 1916-1917, p. 32.

2 See Gluchoff 2011.

and C the ballistic coefficient (corresponding to the form of the projectile). G and H are empirical functions of which tables were available based upon experiments at the proving ground. For H , one could also use an exponential function of the y -coordinate (e^{-ay}) as an approximation.

Moulton's numerical integration of these equations is a variant of finite differences method that works with successive approximations, repeating the same pattern of arithmetic steps until a solution becomes stable. It was a very flexible method: on the one hand, it could be adapted to several classes of situations, on the other hand, it was a more intricately mathematical and hence difficult method for the common artilleryman.

The translation of Moulton's advanced mathematics into simpler arithmetic occupied artillery officers in the early 1920s and they devised step-by-step manuals and example computations that the computers could follow. The following instructions are taken from C.A.R. Hoar's book:

The following tables are used: Logs and $v^2/100$; Table of the G Function; $\text{Log}_{10} H = -0.000045y$.

The following blank forms are used: "Trajectory sheet" Form 5042; "Computing sheet" Form 5041.

x, x', x'', y, y', y'' computed as follows: [...]

On the first line of the trajectory sheet enter the initial values of x, x', y, y' [...] then turn to the small computing sheet and compute the first column for $t=0$. [...]

The process is as follows: Taking the value of x' from the trajectory sheet, look up at the same time $\log x'$ and $x'^2/100$ in the table of logs and squares. Similarly look up $\log y'$ and $y'^2/100$.

Add $x'^2/100$ and $y'^2/100$ to get $v^2/100$. With the latter, enter the G table and take out $\log G$.

With y , as tabulated on the trajectory sheet, enter the H table and take out $\log H$.

Add $\log G, \log H$ and $\text{colog } C$ to get $\log E$.

Add $\log x'$ and $\log E$ to get $\log Ex'$. Add $\log E$ and $\log y'$ to get $\log Ey'$.

Look up Ex' and Ey' from their logs, in the table of logs and squares. [etc.]³

This extract illustrates a number of things. First of all, the computation of a trajectory is very much a clerical kind of work, involving the selection of the right forms that are properly labelled by numbers.⁴ Further, it involves a number of tables, some of a general nature such as logarithm tables, other of a special nature such as the G table that features the function of resistance where values are to be looked up and copied at appropriate moments.⁵ Though some technical mathematical vocabulary is used, the mathematics to be performed is essentially reduced to additions (and possibly subtractions). Finally, the sequence of the simple steps is meticulously described in a way we would nowadays call "algorithmic", as could be performed by a machine. All in all, the form of computation neatly spells out the actions of the computer, s/he is the incumbent of the

³ Hoare 1921, p. 47-48.

⁴ See Agar 2003 on the link between the civil service in the U.K. and mecanisation.

⁵ The "G" of the G -table stands for "Gâvre" the testing ground of the French artillery where these tables were produced, see Aubin 2014.

orderly arrangement of columns and lines of the computing sheet.⁶ The flexibility of Moulton's method provided yet another element of the dispositives of computational practice generated by mathematical procedure. There were two advantages. First, once a standard trajectory had been calculated numerically, trajectories under slightly different conditions could be easily obtained by just one "run" of the same mathematical procedure (through differences and without successive approximation). Second, if the retardation function G changed, it was easy to substitute one table for another in the mathematical procedure. As the artillery officer L.S. Dederick notes during a brief controversy over Moulton's method and its practice in the *Bulletin of the American Mathematical Society*⁷:

Many trajectories are now computed at Aberdeen using more recent data on the resistance function than those of the G table. The advantage of numerical integration here becomes evident. The computer is merely given a new table of the resistance function and he proceeds as before. The physicist does not have to ask the mathematician "can you integrate these new differential equations that my experiments give?"

In other words, the mathematical procedure (and its stepwise translation into computational instructions) is invariant under the exchange of tabular data. The fact that the mathematical procedure remains the same, whether for successive approximation, or for the shortcut of direct computation under slightly different conditions, or even under substitution of the empirical function is a very important property of the numerical integration method.⁸ It allows for a precise and orderly grouping of data in the ballistic problem and its computational treatment.

There are exactly three groups of data: 1) the initial data of the problem, 2) the data pertaining to the empirical function of resistance and drag obtained by experiment, and finally, 3) the mathematical data of the procedure itself.⁹ This last group, the mathematical data, can be obtained through development of the differential equations that correspond to the general problem. The middle group of empirical or experimental data correspond to a function or functions appearing in that differential

6 For a testimony on the time-consuming work of computing trajectories at Aberdeen Proving Ground, see Polachek 1997.

7 Dederick 1928. It responds to a review by J.E. Rowe of Moulton's *Exterior Ballistics* (Bull. AMS vol. 29 p. 229-232) that faulted the book because it could not be used in practice: "it is for the use of the mathematician rather than for the practical ballisticians" (p. 232). A similar debate happened between Hoar and Moulton, see Gluchoff 2011.

8 This property is not to be had with the Siacci-method, in use before numerical integration. The Siacci-method was valid under specific conditions that simplified the situation and had to be complexified considerably if these conditions were not met and higher degree approximations had to be introduced. The advance of weaponry led under the Siacci-theory to triple-entry tables and the use of nomograms, see Alger 1919.

9 Remark that Charles Babbage already had a similar separation of groups of data in his design for the Analytical Machine, distinguishing between data of the operations and numerical data. This distinction is also rooted in the mathematical development of the method of finite differences but is guided by another philosophy, that of the English algebraic school of the early 19th century. See Durand-Richard 1992.

equation (the resistance, the altitude or ballistic coefficient).¹⁰ The group of initial data, finally, correspond to a specific set-up, a specific angle, speed and velocity of a specific artillery instrument, this is the most concrete set of quantities. These three groups of data thus correspond to different levels of generality, but are also precisely situated in the numerical integration method. The mathematical data correspond to the sequence of computation steps. The empirical data are found in the tables used. Lastly, the entry data are the initial data or the new initial data that may be simply inserted into the procedure to directly obtain the next trajectory.¹¹ The fact that they can be rather neatly separated in the procedure will prove to be of importance for later mechanisation.

2. Looking for mechanisation (1933-1943)

During the 1920s and 1930s the officers at Aberdeen Proving Ground pursued the organisation and optimisation of the computations, providing manuals and courses at the Ordnance School designed to be practical in the field, while at the same time they were looking out for other aids for computation. When at the 1929 AMS Meeting Vannevar Bush presented his planned differential analyser, J.W. Barker, reserve officer of the Ordnance, and L.S. Dederick, chief of Aberdeen Proving Ground' Computing Branch, both present at the meeting, immediately talked to Bush about the possibility of using the new machine to do ballistic computations. This conversation was followed up by Dederick's visit to Bush and Caldwell's differential analyzer that was still under construction at MIT in 1931 and a test ballistic computation on the machine in 1932. Finally, it was decided to build a new differential analyzer at Aberdeen Proving Ground in cooperation with the MIT and the Moore School of Pennsylvania. The construction took from 1933 to 1935 and some improvements were gradually added in the years afterwards.¹²

From that time onward, the Ordnance's textbook mentioned "mechanical integration" alongside "numerical integration".¹³ On the positive side, discounting the time needed for the reduction of data and of the machine's set-up, the analyzer worked about fifty times faster than a human computer. On the negative side, mechanical integration did not lend itself to shortcuts, numerical integration's advantage of obtaining slightly different trajectories directly could not be realized on the differential analyzer.¹⁴ But there was a neat localisation of the three groups

¹⁰Because these empirical functions are often "rough" functions, they were often split up in various "zones" where they were more or less "smooth" (continuous). Corresponding to each "zone", different interpolation formulae were used. In practice, these data, strictly belonging to the mathematical data, were put in the tabular empirical data too. The development of better mathematical methods to interpolate and approximate these "rough" functions during the years 1930-1940 constitutes an important chapter in the genesis of numerical analysis. Schoenberg's work, mentioned later in this article is part of that story.

¹¹That is, if they are only slightly different from the preceding initial data.

¹²The information in this paragraph is taken from A.A. Bennett's report (Bennett 1942, p. 7-13).

¹³Kent et al. 1938, p.444

¹⁴See Dederick 1940, p. 634-35 and Feller and Shannon, p. 2.

of data. The differential analyzer being an analogue machine, it did not proceed by simple, discrete computation steps, but rather the quantities were generated by the connection between continuously moving integrators interconnected with so-called input tables. On these input tables one could lay a plotted curve that could be traced using a pointer. On the Aberdeen analyzer, three input tables were used during computation: A divisor table (for doing division), a vector table (that contained the initial conditions in the form of limits on the variable v), and a template table (that contained the G function). Thus, the three groups of data were materially separated: Mathematical data were the cables connecting the integrators and tables, functional data were in the template table(s) and the initial data in the vector table.

This neat separation was challenged by the H function. The H function was not taken from a table, but calculated as an exponential function of y on the differential analyzer. When the Analyzer was up for review by W. Feller and C.E. Shannon in 1942, Dederick asked whether it would not be better to use a table than to calculate it ad hoc, the answer was a clear no (Feller and Shannon 1942, p. 8). The question had already come up at MIT in 1932 when Guerrieri had investigated what functions could be conveniently generated by the differential analyzer itself: "functional relations are introduced into the machine by means of input tables which are not automatic in their operation and which require the services of an attendant for their use [...] the process might possibly be made entirely automatic" (Guerrieri 1932, p.1) Guerrieri had shown how elementary functions (such as the exponential of the H function) could be simply generated by the differential analyzer itself instead of "reading" the data from tables. It soon turned out the idea could be generalized. Indeed, as C.E. Shannon showed in a 1941-publication, essentially all functions could be generated by the differential analyzer itself (Shannon 1941).¹⁵ This results blurs the neat separation between groups of data, since, at least in theory, all functional data could be put in the machine itself instead of in input tables.

With the progress of weapons development during World War II, it was soon clear that the groups of computers and the differential analyzers could not work fast enough to supply all necessary ballistic tables to the army. Also, "the solutions obtained on the Differential Analyzer were not as accurate as those obtained by manual calculation" (Polachek 1997, p. 26). The problem was debated at meetings of the National Defense Research Council (that counted four to five members of Aberdeen Proving Ground). The Council decided it would be wise to supplement the human and analogue computing with digital machinery. George R. Stibitz from Bell Labs had been serving on the NDRC since the beginning of the war and had constructed with his team of engineers at Bell two relay-based digital computers: Model II (or the Relay Interpolator) in 1943 and Model III (or the Ballistic computer) in 1944, both for the Naval Research. In 1944, the NDRC decided that a two new machines of a more general design, the Model V, would be built under contract for the Naval Research and for

¹⁵To be precise, all functions that are not-hypertranscendental. Examples of hypertranscendental functions are the Gamma-function or the Riemann Zeta-function

Aberdeen Proving Ground.¹⁶ In June 1943 they also signed a contract with the Moore School where John Mauchly and Presper Eckert had proposed an ambitious project to build a numerical integrator and computer with vacuum tubes.¹⁷ The Bell machine would finally be delivered December 1946 for the Ordnance and August 1947 at Aberdeen. The ENIAC, Mauchly and Eckert's machine, would be ready February 1946. These machines added to the digital computational power already present at Aberdeen, two IBM Relay calculators that had been operational since 1944.¹⁸

In a report on the computing facilities at Aberdeen in 1949, the authors write, speaking of the IBM calculators, the Bell machine and the ENIAC¹⁹: “The three digital machines have very similar logical organizations” (Harrison et al. 1949, p. 5) and they give the “logical organization of a representative digital calculating machine” as follows:

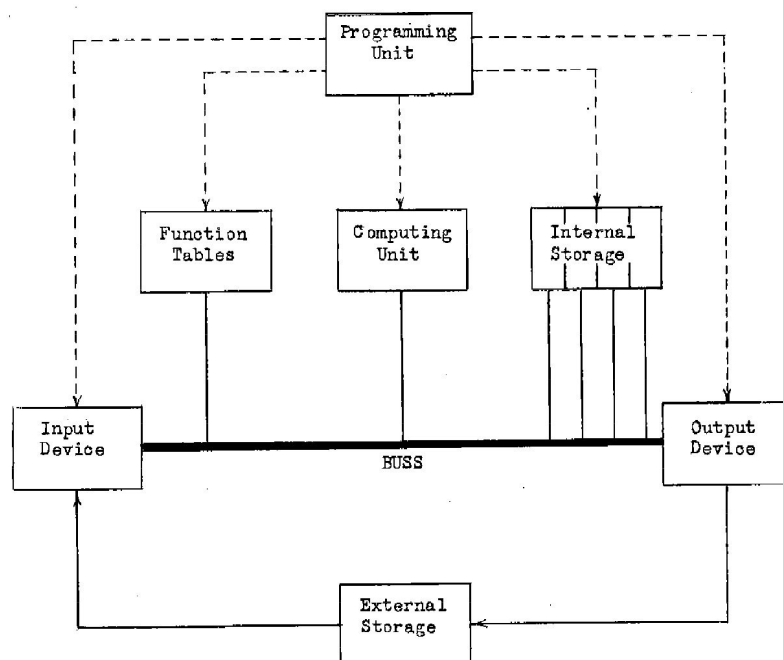


Figure 1 - Block Diagram of the Representative Digital Calculator

In the diagram we have a central programming unit that commands function tables, arithmetic unit and internal storage that all communicate, through a bus, with input and output. It may quite surprise to see a representative model of a computer anno 1949 that does not resemble the well-known von-Neumann-architecture.²⁰ In the von Neumann model, supposed to be the “blueprint” of the modern computer, the logic and arithmetic units are brought together in a central programming unit that

16 See Stibitz' report (March 1944) to the Council, or one of the papers describing the line of Bell relay calculators.

17 See Brainerd 1976 or other papers of the rich bibliography on the ENIAC.

18 These machines are described informally but clearly in Berkeley 1949.

19 The ENIAC the report describes is already the “rewired” ENIAC that “simulates” a stored-program computer, see Haigh et al. 2014 for more details.

20 Von Neumann 1945.

controls communication with the memory device and input/output. But of course, the report presents a synthesis, driven by von Neumann's talent of abstraction and generalization, of how the team that was developing the all-electronic ENIAC was forecasting the form of future computing devices. In contrast, Aberdeen's report is still paying tribute to practices of the past and has to be read in the tradition of the organisation of (ballistic) computations. The difference between von Neumann's and Aberdeen's reports is most obvious in the special unit "Function tables", completely absent in von Neumann's model, but capital in Aberdeen's representative model.

The authors of the report specify in detail:

The internal and external storage furnish the calculator with the analogues of the mathematical symbols of (), [], { }, and other designations of association. [...] In comparing machine computing with manual computing, we may liken the internal storage and external storage respectively to the memory of the [human] computer, and the data sheet upon which he enters intermediate results.

Manual computer have frequent occasion to refer to tables of pre-computed or empirical functions. To perform the analogous operation upon the representative digital computing machine, function tables are provided [that contain] entries and interpolational coefficients of a tabulated function. (Harrison et al. 1949, p. 7)

Also in parallel with the organisation of manual ballistic computation, a division of labour between the mathematician and the computer returns in the form of a relationship between the "problem preparer" and the "machine": "the machine does not solve a problem [...] [it] will not invent the sequence of operations necessary [...] the method of solution must be devised by the problem preparer" (Ibid., p. 8)²¹ The suitability of a problem for machine computation is conditioned by three criteria: The reducibility, the containability and the economy of a problem. A problem is reducible if "the expressions to be evaluated in its execution are expressible to a sufficient degree of approximation, in terms of the elementary operations and pretabulated functions." (p.8) It is containable if it satisfies the digital capacity of the machine (its word size), its storage capacity (its internal and external memory) and its sequencing capacity (the types and organisation of orders of the programming unity). Finally, the economy of a problem must be considered, is the solution method repetitive and general enough as to warrant the investment of the coder's time and the machine's time?

As the comparative table makes clear, the ENIAC stands out for the economy of running time (its electronic speed) and the Bell computer for its unlimited programming capacity and flexibility of changing the set-up (because new data can be automatically accessed).

²¹It may be remarked that this division of labour returns in a slightly different form in Goldstine and von Neumann's report on programming (1947-1948, pp. 20-23).

Conditions of Suitability	Machine Characteristics		
	Eniac	Bell Relay Computer	IBM Relay Calculators
Reducibility	+, -, x, ÷, √ and pretabulated functions		
Containability			
Digital Capacity	10 digits fixed decimal point	7 digits floating decimal pt.	6 or 12 digits fixed decimal point
Internal Storage	20 quantities (200 digits)	15 quantities (105 digits)	20 6-digit quantities 5 12-digit quantities (216 digits)
External Storage	Unlimited (manual access)	Unlimited (automatic access)	Unlimited (manual access)
Programming Capacity	Limited	Unlimited	Limited
Economy			
Coding Time	Several Days		
Testing and Set-Up Time	Several hours	Several hours for 1st set-up zero-time for subsequent set-ups	Several hours
Running Time	.012 sec. per mult.	1 sec. per mult.	.2 sec. per mult.

The common background in ballistic computation for the three digital Aberdeen machines is evident from the language and similes used in this report. It can also be tracked down in the architecture of these machines, but, what is quite interesting, the accumulated experience gathered by using these machines during the years 1946-1950 ultimately led to changing the computational organisation, having an impact on hardware design (abandoning the "representative digital machine" for von-Neumann architecture) and on software conception (from preparing and coding computations to programming). This change can be observed in the development and use of the Bell Model V (especially when compared with the Model III), and in the rewiring of the original ENIAC into a "stored-program" machine.²²

3. Bell Model V Calculator: Tapes and Controls

The Bell relay calculator Model V pursued the line of design of Models II and III²³, but was to be a general-purpose machine from the outset. The head engineer for Model V was S.B. Williams who had already been in charge of the Model III machine. The Model III machine was actually a machine to compute errors, viz. the machine "observed" (i.e. it received the data the communication through punched tape) a real gun director shooting a projectile with a fuse and compared that data with the data of the gun director and the ballistic tables that contained the presupposed trajectory and fuse time of the projectile. The difference between these data gave the errors. This set-up is translated into three types of tapes

²² See references of footnote 19.

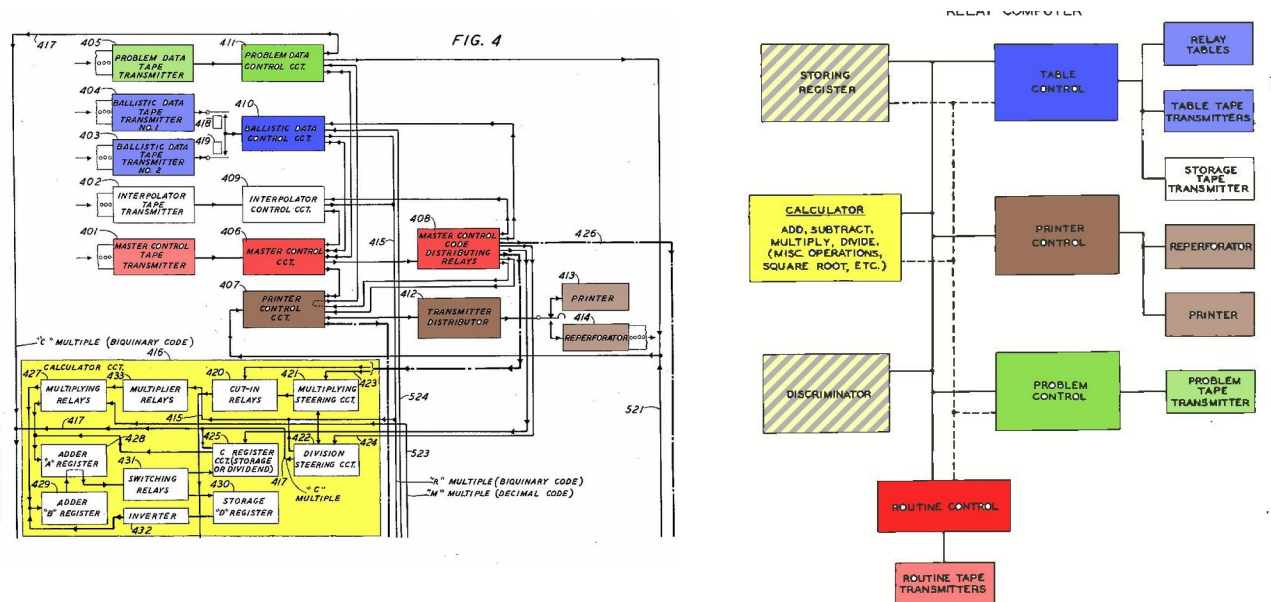
²³ Model IV was a slightly different version of Model III, with extra trigonometric and logarithmic tables that had been wired down.

read by the Model III machine: Ballistic tapes containing known and precomputed ballistic data; the interpolator data tapes that contains the mathematical coefficients for the interpolation; and the problem data tape containing the data “observed”. Finally, there was the master or routine tape. As the patent specifies:

The master tape will bear the burden of steering the calculations, the tapes which have herein been called the ballistic tapes will contain a great amount of known information tabulated by indices (which may be calculated or recorded by this calculator), the problem data tape will contain the arguments of the problem or problems to be solved and the interpolator tape will contain data correlated with the problem data.²⁴

On the Model III the problem data came from an external source (the “observed” projectile), so the machine was essentially in a feedback loop with another piece of machinery, it was a coupled system. Though the engineers sometimes suggested Model III was nearly general-purpose, its coupling with the gun director made it special-purpose. This was the main difference in design of Model V that would be a standalone computer, not coupled to external devices like a gun director, and built as a general-purpose machine from the start.

This transpires from the respective block diagrams of the relay computers. Model III is basically oriented from input (5 tapes²⁵) to output (1 tape printed), where the data are first combined and then transformed under way by the calculator (in yellow in the figure). Model V is self-contained and is governed by the routine tape (in red) that controls the communication with and between the other units.



The block diagrams of Models III and V make the consequences of this transition visible though they also show some continuity. On the left hand

24 Andrews et al. (1946) p.24-25.

25 These are : the control data tape (red), the interpolation data tape (white), the two table data tapes (blue) and the problem data tape (green). The tape recording the output is brown on the figure.

side of Model V's diagram two additions are featured (diagonally striped), the discriminator and the storage register. On the right hand side, we have virtually the same block elements, the tapes and their controls, as in Model III, though they have been slightly rearranged. The left hand additions are symptomatic of the growing importance of "programming", controlling the routine. The right hand side rather marks the continuity with the older practices of automated ballistic computation.

In Model V, the disappearance of the interpolator tape and control (white) is immediately linked to the appearance of the discriminator unity (striped), they are effects of moving away from a specific class of problems (ballistic equations and interpolations) to a more general and abstract approach to mathematical computation. As mentioned earlier (footnote 10), the ballistic calculations used empirical functions (the drag function G) that were rather "rough" so they had to be approximated and interpolated locally. Instead of a global interpolation formulae, a set of local interpolation formulae with specific coefficients were matched to the different "zones" of the "rough" function. These coefficients and the addresses of the "zones" where they were to be used could be found on the interpolator tape. On the original ballistic machines, Models II and III, the machine had a special control coupled to the interpolator tape to switch between interpolation formulae. This specific setup was now abandoned for a more general formulation. The discriminator unit could now be used to look at specific values provided by the computation and to "discriminate" or "decide" on the basis of that value what would be the (sub)routine to be followed, viz. to activate parts of the routine tape and/or data in storage to pursue the calculation. This change in machine design features two important characteristics of programmability: Abandoning special-purpose for general-purpose, and the introduction of a conditional structure in the computation.

Actual usage of the Model V machine at Aberdeen, however, soon made clear that this logical design of the machine could have even been improved upon. In particular, the distinction between the various tapes, still inherited from the organisation of manual ballistic computation, proved obsolete:

When the tape input system for this machine was designed, it was intended that the routine tapes should contain all the orders, the table tapes should contain numerical information of a general nature, comparable to function tables used in manual computing, and the problem tape should contain numerical information specific to the problem being solved (i.e. initial conditions for differential equations, coefficients of a system of algebraic equations etc.). Gradually, as the machine developed, the distinction between the kind of tapes became blurred. As matters stand now, most orders are contained in the routine tapes, but some may be put in the problem tape; some numerical information may be put into routine tapes, and most of it is distributed between the table and problem tapes at will, to fit the needs of a particular problem. [...] As a result of the experience gained in the operation of this machine, it appears that the problem tape is an unnecessary complication of the machine, and that its

function had better be distributed between the other two types of tapes. (Alt 1948, p.9)

The logic and memory management of a manual computation indeed suggest a distinction between problem data and tabular data. The problem data are the data you start with and that help select the mathematical procedure you will use, the tabular data are consulted as the computation develops and demands those data. Moreover, the problem data are used only once and then exchanged for new initial data, whereas the tabular data are permanently stored in printed books for repeated usage. This manual distinction between one-time-usage and repeating usage is superseded by the machine distinction between fast or slow access to data. In a general-purpose computing device as Model V you essentially work with a few relatively fast memory registers that are used to hold the values that you are working on and that you continually need. To add or multiply in the registers takes 0.3 and 1 second respectively. If other data, problem data or tabular data, are needed, values from the tapes can be accessed, rather slowly in the range of some seconds to some minutes²⁶, and transferred to the computing registers. This is the relevant distinction on the relay-based Model V: There is a routine and the values it is handling, and there is a table and values that can be looked up when needed. The first can be accessed rapidly, the second rather more slowly. The distinction between fast and slow data will become the all-determining characteristic of the all-electronic vacuum-tube-based ENIAC.

4. *The ENIAC or how electronic speed prompted automated programmability*

The engineers at the Moore School had gathered know-how with the construction and use of the differential analyzer since 1935. Further, there had been teams of computers at the Moore School, and in 1942, John Mauchly had learnt to organize such a computers' team for doing numerical integration (Aker 2008, p. 85). Mauchly's central idea was to exploit the speed of vacuum tubes for digital computation and together with the engineer Eckert they thought they could manage to turn the notoriously unreliable tubes into a reliable digital adding machine (or accumulator). Mauchly had written a memo on how to use vacuum tubes for ballistic calculations in 1942, essentially using the organisation of a manually performed numerical integration as the blueprint of the architecture of the proposed machine. In 1943, this memo was the basis for the proposal to the NDRC, but now the expertise of Eckert and Brainerd brought in elements of the organization of a differential analyzer. In a way, the ENIAC is the fruit of this confluence of ideas, an "amalgam between

²⁶A random entry on a table tape can be found (can be "hunted" as the Bell jargon had it) in under 8 minutes, in most cases, the entries used are "close" to one another, so the access time is rather in the bound of some seconds. The problem and routine tapes are read on average at the speed of one number or instruction per 2 seconds. Because access to the tapes was relatively slow, the frequently used classical functions such as sines or logarithms were "wired down" as relay circuits: "permanent function tables in the machine, [t]hese are wirings which make it possible to obtain very quickly the value of certain functions for any desired value of the argument (sin, cos, tan-1 and log" (Alt 1948, p. 74)

the differential analyzer and the knowledge of human computers that became embedded into the ENIAC's proposed design" (Aker 2008, p. 90). The hardware development of the ENIAC happening between the original proposal from 1943 and the extensive 1946-descriptions of the machine as it was finally built, was a long and complex process.²⁷ One thing was to get a decade ring counter for storing and processing numerical values up and running, and later, to get a complete accumulator (an adding machine) made out of vacuum tubes to work reliably. Yet another thing was to develop the units of the ENIAC that featured more or less as "black boxes" in the 1943 proposal: the central programming unit and the function generator. The programming unit had to tie the workings of all accumulators together, and the function generator was meant to provide functional data to the calculation (as would a table in manual calculation or in a differential analyzer). As was often the case in circuit design, design problems often first appeared as "black boxes" in the overall picture, to be filled in later on in the design process.²⁸

For practical, engineering reasons²⁹ the "local programming" of the accumulators was done first, end of 1943 (Aker 2008, p.99). This "local" programming essentially defines how the communication with the other units would run. It tells when a result has to be "written" in another column, or has to be added/subtracted n times from another value etc. The master programmer that would control the sequence and repetition of such elementary "local programs" was only designed mid 1944. Arthur Burks, one of the engineers on the ENIAC project, was responsible for a large part of its design. In 1944, he had made a schematic for how the projected machine would compute ballistic trajectories using the Heun method³⁰. This would serve as a blueprint when they started developing parts of the machine. The master programmer, responsible for the "central programming".

The "function generator" was defined as "a device for introducing an arbitrary function, the mathematical form of which is either not known or not simple enough to be generated by a simple difference equation." (Brainerd et al. 1943, Appendix 2, p. xi) The generator would read and

27 For an overview of the design process see p. 85-102 in Aker 2008 and Burks 1965.

Apart from the know-how the engineers at the Moore School, there was also interaction with the Bell Labs engineers (S.B. Williams acted as a consultant and G.R. Stibitz was member of the NDRC panel, though the extent of the Bell contribution is unclear) and with many other people from the NDRC panel and Aberdeen Proving Ground

28 Cf. "The correct approach to multifunctional circuit design is to some extent the inverse of the analytical operation. From a statement of circuit requirements, a functional plan is developed in terms of known or conceptually evident circuit blocks, representing simple circuits similar to single-function circuits [...] as the design proceeds, the functional blocks are coordinated and integrated to the point where a comprehensive block diagram of the proposed circuit exists. [...] The most satisfactory approach to developing a block diagram is to start with a few main subdivisions of the over-all circuit and successively break these down until each block represents a unifunctional circuit." (Keister et al. 1951)

29 Viz. they first built two accumulators, connected and programmed them to show the feasibility of the machine.

30 According to Mauchly, he came up with a method simple enough to be put on the ENIAC, and this method was later on identified as a method developed by Heun (Stern 1977).

store information from punched cards “permitting it to continue generating the function for a limited range in x ”, and, “the function generator in effect interpolates a great number of values of $f(x)$ when provided with $f(x)$ and some of its differences at relatively few points.” (Idem, p. xii) The idea was that the argument x would cause not only $f(x)$ to be looked up and transmitted, but also upto four surrounding values $f(x-2)$, $f(x-1)$, $f(x+1)$, $f(x+2)$, all could be transmitted upto 9 times, possibly with opposite sign. This arrangement was helpful for building interpolation formulae in the accumulators.³¹ The team took their inspiration from a function generator design developed at RCA (Morton et al. 1944) that they had already used to devise a multiplication table for the ENIAC's multiplier. However, the RCA design, though it had worked for the 10x10 multiplication table, proved impracticable for the function table that would need 104 entries times 12 digits and 2 signs (with the flexibility of reaching for 4 neighbouring values). The design was adapted mid 1944 by Burks and Shaw (Burks 1964, p. 36-44).

The ENIAC conditioned the kind of mathematics that could be used for the ballistic trajectories by its limited programming and memory resources and by its phenomenal speed. The ENIAC could do 5000 additions per second and it could consult about 1000 values from its function table per second, but reading a punched card took half a second and crippled its speed. This was the reason the Heun method was preferred over methods that had been used by human computers. Simplicity of method and brute force computation with vacuum tubes won over more subtle, but less straightforward methods of calculation. But the ENIAC, in combination with weapons development and improved measuring techniques, also challenged mathematics. New classes of G functions had been introduced in the 1930ies (G1 to G5), after 1943 there were even more:

All firing tables prepared by the Laboratory before 1943 depended on a limited number of standard drag functions and drag coefficients which had been worked out for generic types of projectiles between the wars. Whenever a drag function was needed for a projectile of new design, the function already established for the type of projectile most closely resembling the new design was used. By 1943, however, the use of solenoid chronographs in conjunction with electronic counters made possible the rapid and accurate calculation of drag functions for new projectiles on a fully experimental basis. (Barber et al., p. 45)

These new drag functions only had a limited accuracy of 4 digits and it was feared that, after the many Heun iterations, it could generate non-negligible rounding-off errors. Also, the “rough” contours of the G-functions had to be handled with care and it seemed that new approaches

³¹ For instance, to obtain $f(67,320)$ by simple linear interpolation between two known and tabulated values, say $f(67)=72500$ and $f(68)=74500$. The function table is set to transmit not only two times the value of $f(67)$ (i.e. the value corresponding to the first two digits of 67,320) but also to transmit one the next value in the table, $f(68)$, with negative sign. The remaining digits (320) are used as coefficients in the interpolation and are also transmitted this time to the 2nd multiplicand unit of the multiplier. The difference between $f(67)$ and $f(68)$ is taken in one accumulator (the 1st multiplicand unit of the multiplier), then multiplied by 0,320, the result (0.640) is added to $f(67)$ (in another accumulator) to obtain 73,140. The example is taken from Eckert and Mauchly 1947, p. 66.

to interpolation, either of higher degree than four or of a greater variety, had to come in. These problems were attacked in two ways: Locally, the ENIAC team exploited the hardware to come with robust solutions, whereas the Aberdeen Research Lab put mathematicians to work to find mathematical solutions.

In 1945-46, the ENIAC team had already devised a scheme that could work around some of the problems. In this scheme, the function table needed not only to be a look-up table linked up with some interpolational coefficients, but had to incorporate a way of discriminating what “zone” of the table had to be used (Report ENIAC, vol. 4, VII, 34-35).³² Indeed the table of the drag function could not be simply interpolated with just one formula, but had to be approached in “slices”. As can be seen from the form of a typical G function (taken from Reed 1952, p. 103), the curve consists of about three “zones”. Before 1000 on the X-axis (i.e. when velocity becomes faster than sound), the function is slowly increasing, but then, for a short while, steeply goes up, to restabilize and increase steadily but slowly.

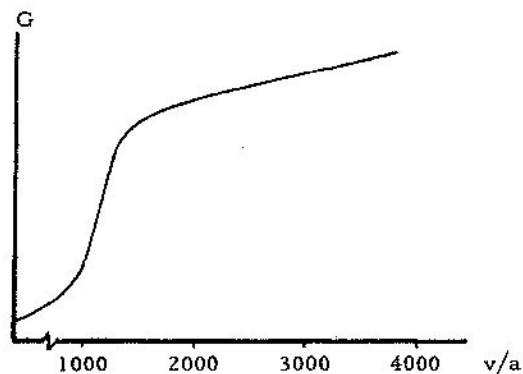


Figure 1

It was therefore proposed to modify some of the digits that are not used in the function table (because the accuracy of the G-function does not go to 12 digits). Depending on the value V with which the function table is entered (viz. velocity smaller than velocity of sound, in its neighbourhood or greater than) the 9th, 10th or 11th digit will be 9 (the columns are marked grey in the diagram below). This digit will then be rerouted to a so-called “dummy program” of an accumulator. This “dummy program” essentially “translates” the numerical 9-pulse into a programming pulse (B(11), B(10) and B(9) respectively) that will then stimulate the appropriate interpolation routine and will send the argument a either to function table 1 (for “zones” 1 and 3) or function table 2 (for “zone” 2 in the vicinity of the velocity of sound), as can be seen in the block diagram below.

³²The author of Parts 3 and 4 of the ENIAC report is Adèle Goldstine, the wife of H.H. Goldstine and on the ENIAC's programming team, although she most probably served rather as the editor of the report, gathering technical reports and facts of the ENIAC project and compiling the «Technical Description». A specialist of ballistic computations herself, she may, however, have contributed directly to this scheme.

		B(1) - Routine Table 1									
ARG.		A(17)	A(16)	A(15)	A(14)	A(13)	A(12)	A(11)	B(9)	B(10)	B(11)
a'	G(v ²)								0	0	9
									0	0	9
a''	G(v ²)								0	9	0
									0	9	0
									9	0	0
									9	0	0

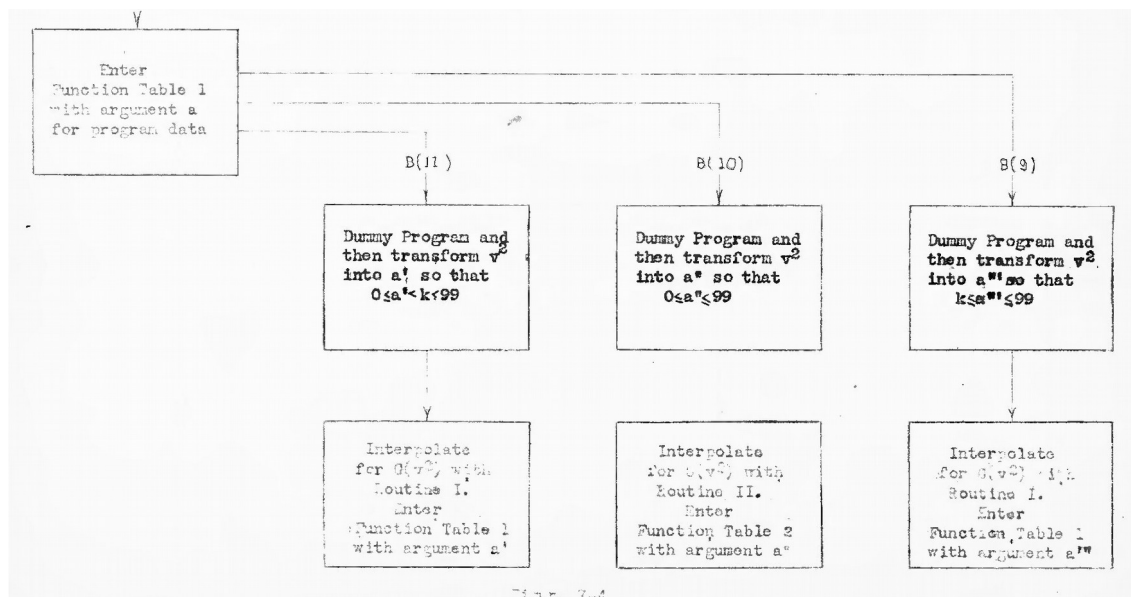


Fig. 7-4

The function table is thus modified to generate programming pulses next to numerical pulses. In quite a similar way as for the Bell machine, the boundaries between numerical data and programming data also became blurred for the ENIAC. This set-up can even be extended and the whole function table could eventually be modified into a "plugboard" for stimulating programs (i.e. subroutines). This idea would prove to be very fruitful (and the subject of much controversy). When, after much discussions and interactions between the ENIAC team, the programmers, the mathematicians, ballistic officers and John von Neumann, it was decided to "rewire" the ENIAC in a permanent way, the function table

would be used to as the pathway to stimulate the basic orders of the ENIAC. In this way, the ENIAC, though admittedly slower in its operation, became much easier to program now it could “simulate” a stored-program computer and sequentially execute orders.³³

Taking a look at the mathematical work on the ballistics for the ENIAC, it was started in 1943 when Hans Rademacher was hired to look at the rounding-off problem, and when I.J. Schoenberg was asked to develop methods to “smoothen” the rough G functions:

Trajectories of projectiles were until then computed with desk calculators by hand. Into these computations entered tables of the drag-functions of air resistance, about 24 of them, which were obtained empirically by firings of various types of projectiles. As the step of integration used in these trajectory computations was rather large and the methods of numerical integrations fairly complicated, it did not much matter that the 4-place drag-function tables were rather rough. In performing these computations on the ENIAC, which was very fast, a much simpler integration method of very small step could be used. In these methods, the accumulation of the round-off errors was unacceptable due to the rough drag-function tables; they needed to be smoothed by being approximated by analytic functions. To do this was my problem. (Schoenberg 1988, p.4)

Schoenberg came up with a variety of methods that were known as osculatory interpolation and that he would later systematize and generalize under the name of “splines”. As for Rademacher, his work on the accumulation of rounding-off errors (Rademacher 1948) would also be one of the pioneering studies in modern numerical analysis. Rademacher's results helped determine how many digits each accumulator should have for the desired precision, viz. 10 (decimal) digits (Haigh et al. 2016, p. 34 & 294).

Schoenberg's work on interpolation was implemented in the ballistic calculations after 1947-1948. In a description of the standard calculations, Harry Reed, one of the operators for ENIAC's ballistics computations, writes:

The method of integration used to compute trajectories on the Eniac is one due to Heun. [...] The error in this method is of the third order in Δt .³⁴ Other methods using higher order approximations have been considered, but the above is quite well adapted to the Eniac's storage capacity and has sufficient accuracy for the allowable error under suitable choice of Δt . (Reed 1952, p. 104)

The remark that the Heun method is “quite well adapted” is an understatement given that the Eniac was in part designed with this particular method in mind. Schoenberg's contributions appear in the part where interpolation from the G-table has to be done. According to the “quadrant angle of elevation” being smaller or greater than 45 degrees a different order of interpolation is chosen. In the first case, osculatory

³³See Clippinger's report and the articles by Neukom for more technical details on this complex reworking, and see Haigh, Priestley and Rope 2016 (Chapter 7) for more details on this complicated story.

³⁴This was the main result of Rademacher's study.

interpolation for y is followed by a second interpolation to obtain the angle using Aitken's algorithm. In the second case, osculatory interpolation is used to obtain x and then Aitken's algorithm is used for obtaining the angle. Reed remarks: "there seems to be difficulty in obtaining smooth results therewith, and the use of osculating interpolation appears to be the better approach" (Reed 1952, p. 105).

Conclusion

The particular intertwining between mathematical methods and computational organization in ballistics that was developed and taught at Aberdeen Proving Ground had quite an impact on the decision-making and on the design of analogue and digital computing instruments that were built between 1935 and 1947. The division of labor inherent in Moulton's reorganization of ballistic computing (the procedure to be followed, the tables for look-up and the simple but repetitive arithmetic operations for the human computer) proved to be the initial blueprint for the digital machines. The handling of the H function and in particular of the G function proved to be tricky. Already on the differential analyzer it was clear that the H function could be better generated than tabulated. When the Bell Model V and the ENIAC were built, this example was followed and the H function was directly generated.

Because of the advance in measuring techniques, the increasing complexity of the G functions, demanding the regimenting of more than one interpolation procedure, proved to be a bigger problem for the designers of the Bell machine and the ENIAC. In both cases, hardware solutions were worked out. For the Bell machine a discriminator was introduced, on the ENIAC the function table was slightly diverted from its original aim (by using the adapter PX-4-119) to accommodate not only numerical but also programming signals. In a way, one could say that the tables gradually dissolved into the machine. These hardware changes heralded a more general development, that of programmability. Consulting a card punched or paper taped table took too long in comparison with the speed of computation. Putting the tables into fast storage (the extra registers on the Bell Model V and the accumulators and the function table on the ENIAC) proved to be too expensive, because this fast storage was sorely needed for exploiting the speed and versatility of these machines. The special-purpose machines grew to become general-purpose. The key was programmability. To fully exploit the speed of the machines, the program had to be internalized and read at high speed so that the machine could compute autonomously, without waiting (too much) for external input.

With the progress of technology, both of weapons and of vacuum tubes' electronic speed, new mathematical techniques had to be brought in too. F.R. Moulton's numerical integration technique had started off the "scientification" of the U.S. Ballistic research and had prepared the minds for mechanization. But conditioned by the materiality of the vacuum tube machine, its enormous speed coupled with a very limited fast memory, other mathematical procedures were needed. Numerical integration was (temporarily) abandoned for a less advanced but more straightforward

technique, Heun's method. Not only the main algorithm for ballistic calculation changed, but also “new”³⁵ problems appeared: The study of rounding-off errors and the development of new classes of interpolation methods.

This two-way development of mathematics and computing machines did not go unremarked, already John Mauchly, after Rademacher's talk on rounding-off errors during the Moore School Lectures in 1946, said: “many of us have been looking forward with certainty to the day when the advent of computing machines would influence the course of mathematics, and there are still many directions in which we can look forward to further such impact. This reaction is not, of course, one way. Mathematics should also influence the computing machine.” (Moore School Lectures 1946, p. 186). While the computer certainly helped to explore and develop new mathematic problems questions, and in particular contributed to the establishment of a new flourishing mathematical discipline, viz. numerical analysis, the impact of mathematics on machine design has diminished from the 1950s onwards. Through the development of programming systems in the 1950s the « trading zone » between man and computer migrated from the machine and its components to a variety of programmable interfaces that could be more readily and flexibly developed and modified according to the human user's needs. As a result, the interaction between mathematics and computing moved to the « software » side of things. Mathematics and its formal expressiveness served as the reference point for the first programming languages such as Fortran (1957) or Algol (1960), and in its turn, problems in developing programming tools led to new directions of mathematical research such as automata theory or formal verification. As computing progresses and moves on, so does its interaction with mathematics.

BIBLIOGRAPHY

- A Report on the ENIAC*, 1946, 5 volumes: vol.1 ENIAC Operating Manual (A.W. Burks and H. Huskey); vol. 2 Maintenance Manual (Chu, Cummings, Davis, Huskey, Sharpless, Shaw); Vol. 3-5 Technical Description Parts I (A.K. Goldstine) and II (H. Huskey), Moore School of Engineering.
- National Research Council, *Annual Report 1916-1917*, Washington D.C.
- AGAR (John), 2003, *The Government Machine. A Revolutionary History of the Computer*, Cambridge/MA, MIT Press.
- AKERA (Atsushi), 2008, *Calculating a Natural World: Scientists, Engineers, and Computers during the Rise of U.S. Cold War Research*, Cambridge/MA, MIT Press.
- ALGER (P.L.), 1919, « Charts for the calculation of the effect of small changes in the elements of fire », *Journal of the United States artillery*,

35 “New” is very relative in this context. The problem of rounding-off was well-known in the 18th and 19th centuries, mostly in an astronomical context. Similarly, Schoenberg found inspiration for his interpolation methods in actuarial mathematics from the interwar period. Rather the digital electronic computer reinvigorated interest in fields of mathematics and its applications that were either marginal or somewhat forgotten.

- 51, pp 585-603.
- ALT (F.L.), 1948, « Bell Telephone Laboratories' computing machine »», *Mathematical Tables and Other Aids to Computation*, 3, p.1-13 & 69-84.
- ANDREWS (E.G.) and VIBBARD, (V.L.), 1946, *Patent application Automatic Calculator*, filed Dec. 17 1946, granted March 28 1961, no 2,977,048.
- AUBIN (D.), 2014, « "I'm Just a Mathematician": Why and How Mathematicians Collaborated with Military Ballisticians at Gâvre »», in Aubin (D.) and Goldstein (C.), édcs., 2014, *The War of Guns and Mathematics: Mathematical Practices and Communities through World War I in France and its Western Allies*, Berlin, Springer.
- BARBER (G.) et al., *Ballisticians at war and peace, A History of the United States Ballistic Research Laboratories, volume 1: 1914-1956*, Aberdeen Proving Ground, Maryland.
- BENNETT, (A.A.), 1942, *Report on the Differential Analyzer at Aberdeen Proving Ground, Ballistic Research Laboratories*, Report 319, Aberdeen Proving Ground, Maryland.
- BERKELEY (E.C.), 1949, *Giant Brains. Or machines that think*, Science editions, New York.
- BLISS (G.A.), 1919, « A method of computing differential corrections for a trajectory » , *Journal of the United States artillery*, 51, p. 445-449.
- BRAINERD (J.G.), ECKERT (J.P.), MAUCHLY (J.W.), 1943, *Report on an Electronic Diff* Analyzer, for Ballistic Research Laboratory*, Aberdeen Proving Ground, by Moore School of Electrical Engineering, University of Pennsylvania, Unpublished, April 1943. (contains Appendices A, B and C)
- BRAINERD (J.G.), 1976, « Genesis of the ENIAC », *Technology and Culture*, Vol. 17, No. 3, p. 482-488.
- BURKS (A.), 1965, *Exhibit A of Affidavit of Arthur W. Burks*, Honeywell vs. Sperry Rand, Consolidated Civil Action No. 4 - 67 Civ. 128, US District Court, Minnesota Fourth Division.
- BUSH (V.), 1931, « The differential analyzer. A new machine for solving differential equations », *Journal of the Franklin Institute*, 212 (4), pp 447-488..
- CALDWELL (S.H.), 1933, *The extension and application of differential analyzer technique in the solution of ordinary differential equations*. PhD Thesis MIT.
- CAMPBELL-KELLY (M.) and WILLIAMS (M.R., eds.), 1985, *The Moore School Lectures: Theory and Techniques for Design of Electronic Digital Computers*. Cambridge, Massachusetts; London, England; Los Angeles, California; San Francisco, California, The MIT Press.
- DEDERICK (L.S.), 1928, « Letter to the editor » , *Bulletin of the AMS*, vol. 35, p. 667.
- DEDERICK (L.S.), 1940, « The mathematics of exterior ballistic computations », *American Mathematical Monthly*, 47 (9), p. 626-634.
- DURAND-RICHARD (M.-J.), 1992, « Charles Babbage (1791-1871) : de l'Ecole algébrique anglaise à la "machine analytique" », *Mathématiques, Informatique et Sciences Humaines*, 30° année, n° 118, p. 5-31 et n° 120, p. 79-82.
- ECKERT (J.P.), MAUCHLY (J.W.), GOLDSTINE (H.H.), BRAINERD (J.G.), 1945, *Description of the ENIAC and comments on electronic digital computing*

machines, NDRC AMP Report 171.28 November 1945.

ECKERT (J.P.) and MAUCHLY (J.W.), 1947, *Electronic Numerical Integrator and Computer*, filed June 26, 1947, patented Feb. 4, 1964, US Patent no 3,120,606.

FELLER (W.) and SHANNON (C.E.), 1943, *On the integration of the ballistic equations on the Aberdeen Analyzer*, Applied Mathematics Panel Report no. 28.1, National Defense Research Committee, in: Sloane (N.J. A.) and Wyner (A.D., eds.), 2013, *C.E. Shannon. Miscellaneous Writings*, Bell Labs.

GOLDSTINE (H.H.), 1977, *The Computer - From Pascal to von Neumann*, Princeton, Princeton University Press.

GOLDSTINE (H.H.) & NEUMANN (J. v.), 1947-1948, *Planning and Coding of Problems for an Electronic Computing Instrument. Report on the mathematical and logical aspects of an electronic computing instrument* (Volume II, parts 1, 2 & 3). Princeton NJ : Institute for Advanced Study.

GLUCHOFF (A.), 2011, « Artillerymen and mathematicians: Forest Ray Moulton and changes in American exterior ballistics, 1885-1934 », *Historia mathematica* 38, p. 506-547.

GRIER (D.A.), 2005, *When computers were human*, Princeton: Princeton University Press.

GUERRIERI (J.), 1932, *Methods of introducing functional relations automatically on the differential analyser*. PhD Thesis MIT.

HAIGH (Tom), PRIESTLEY (Mark) and ROPE (Crispin), 2014a, « Reconsidering the Stored Program Concept », *IEEE Annals for the History of Computing*, 36 (1).

HAIGH (Tom), PRIESTLEY (Mark) and ROPE (Crispin), 2014b, « Engineering 'The Miracle of the ENIAC': Implementing the Modern Code Paradigm » *IEEE Annals for the History of Computing*, 36 (2).

HAIGH (TOM), PRIESTLEY (MARK) AND ROPE (CRISPIN), 2016, *Eniac in Action. Making and remaking the modern Computer*. Cambridge (MA) : MIT Press.

HARRISON (J.O.), HOLBERTON (J.V.) and LOTKIN (M.), 1949, *Preparation of problems for the BRL calculating machines*, Ballistic Research Laboratories, Technical Note 104, Aberdeen Proving Ground, Maryland.

KENT (R.H.), DEDERICK (L.S.) and ZORNIG (H.H.), 1938, Chapter « Exterior Ballistics », in: Hayes (T.J., ed.), *Elements of ordnance; a textbook for use of cadets of the United States Military Academy*, New York: Wiley, p.379-468.

HOAR (C.R.), 1921, *A course in Exterior Ballistics*. Ordnance Textbook, Washington.

Juley (J.), 1947, « The Ballistic Computer », *Bell Laboratories Record*, 24, p. 5-9.

KEISTER (A.), RITCHIE (D.) and WASHBURN (S.), 1951, *The Design of Switching Circuits*. New York, Van Nostrand.

LYNCH (J.), 1947 *Differential Analyzer - electrical aspects of operation*, Ballistic Research Laboratories, Report 654, Aberdeen Proving Ground, Maryland.

LYNCH, (J.) et al., 1949, *Programming principles for the IBM relay calculators*, Ballistic Research Laboratories, Report 705, Aberdeen Proving Ground, Maryland.

- MORTON (G.A.) et al., 1944, *Computing device*, patent filed Jan, 5 1944, granted Feb. 10 1948, US Patent 2,435,841.
- MOULTON (F.R.), 1919, « Numerical integration of differential equations »» *Journal of the United States artillery*, 51, p. 40-55.
- NEUKOM (H.) The Second Life of ENIAC, *IEEE Annals for the history of computing* 28 (2006), p. 4-16.
- NEUMANN (J. v.), 1945, *First Draft of a Report on the EDVAC*, Report Contract No. W-670-ORD-4926.
- POLACHEK (H.), 1997, « Before the ENIAC », *IEEE Annals for the history of computing*, 19 (1997), p. 25-30.
- RADEMACHER (H.), 1948, « On the accumulation of errors in processes of integration on high-speed calculating machines, » in: *Proceedings of a symposium on large-scale digital calculating machinery*, Harvard University. (An earlier version of the paper, "On the Accumulation of Errors in Numerical Integration on the ENIAC" (July 22, 1946)" can be found in the Moore School Lectures)
- REED (H.), 1952, « Firing table computations on the ENIAC », » (Pittsburgh) p. 103-106.
- SCHOENBERG (I.J.), 1988, « A brief account of my life and work », in: *Collected papers, vol. I*, ed. By C. de Boor, Basel, Birkhäuser.
- SHANNON (C.E.), 1941, « Mathematical Theory of the Differential Analyzer », *Journal of Mathematics and Physics*, 20 (4), 337-354.
- STERN (N.), 1977, Interview with Dr. John W. Mauchly, Ambler, Pennsylvania , May 6, 1977 , American Institute of Physics Oral Histories Transcripts.
- STIBITZ (G.R.), 1944, *Calculating System*, Bell Labs Report 29 March 1944. (kindly provided through the courtesy of Crispin Rope and Thomas Haigh)