'A pretence of what is not'? A study of simulation(s) from the ENIAC perspective.

Liesbeth de Mol

`A pretence of what is not'? A study of simulation(s) from the ENIAC perspective.[1]

Liesbeth De Mol, CNRS, UMR 8163 Savoirs, Textes, Langage[2]

"*The past went that-a-way. When faced with a totally new situation, we tend always to attach ourselves to the objects, to the flavor of the most recent past. We look at the present through a rear-view mirror. We march backward into the future*."

(Marshall MacLuhan 1967)

1. INTRODUCTION. `ES GIBT KEINE SIMULATION'

What, if anything, is the impact of technology *x* on science *y*? While instances of this kind of questions have long been neglected within more traditional history and philosophy of science, the steady reversal of the primacy in the science-technology relation since the 1980s (Forman 2007) has given these questions new relevance and put them into a methodological framework where the focus is much more on how certain technological advances affect and shape scientific practice and the knowledge it produces. This focus on technology is, in itself, a historical phenomenon and partially rooted in the growing dependencies between certain branches of science and increasingly complex, and often expensive, technologies where the latter require an expertise which is not necessarily that of the scientist or team of scientists using the technology.

One important driving factor in this development was the possibility of high-speed computation. Indeed, today most, if not all, such technological complexes require intricate computational set-ups which are used in a way which was simply not possible before the development of the high-speed computer. It is from that context that one can understand the recent attention for so-called computational science and the changing role and identity of computer science in the disciplinary spectrum from a discipline struggling for independence from mathematics and engineering to one which has by some been identified as an entirely new scientific domain on a par with the life, physical and social sciences (Tedre 2015; Rosenbloom 2015). Thus, one important instance of the above question is:

---

1 Part of this work is based on a talk given at the conference *The Plurality of Numerical Methods in Computer Simulations and their Philosophical Analysis,* 3-4 November 2011, titled *ENIAC, matrix of numerical simulation(s).* The talk was prepared by M. Bullynck, M. Carlé and myself and dedicated to the memory of Friedrich Kittler.

2 This paper is to some extent determined by some of the ideas elaborated in the ANR project PROGRAMme (ANR-17-CE38-0003-01). I am indebted to Maarten Bullynck, Martin Carlé and Mark Priestley for discussing aspects of this paper. Special thanks go to Sibylle Anderl, Arianna Borrelli, Nathalie Bredella, Rudi Seising and Janina Wellmann for their extensive comments on an earlier version of this paper.

What, if anything, is the effect of high-speed computation on science y?

The current historical (and philosophical) literature which deals with this question focuses for the most part on one particular aspect of that question, viz. the impact of *computer simulation* (Hashagen 2013).[3] One of the standard historical references in this context is Galison's work on Monte Carlo methods and how it required as a simulation method the constitution of a so-called *trading zone* where different practices (including technological and scientific practices) are locally coordinated by relying on a so-called pidgin language (Galison 1997; Galison 2011). But while Galison's work is surely an important contribution to the field, it is in need of a revision (See also (Borelli forthcoming)). Indeed, since Galison, hardly any work has been done to reconsider this almost mythical beginning of computer simulation in the ENIAC context.

In the meantime, in the more philosophical and epistemological literature, it has become clear that *simulation* does not have one stable meaning but covers a number of understandings that give different answers to the question:

What, if anything, is the impact of (computer) simulation on science y?

For instance, in (Duran, forthcoming)[4] it is argued that there are two basic interpretations of computer simulation that co-exist not just within the current philosophical literature but throughout the (short) history of computer simulations. These are:

1. Simulations as problem-solving techniques: this viewpoint comes down to the idea of using computer simulations to solve a set of mathematical equations or, in short, to implement the model. It is here that one can situate the idea of simulations as models in time or dynamical models for short (Cfr Hartmann 1996). In this framework, the idea of one process which imitates another plays a central role.

2. *Simulations as descriptions of patterns:* under this view a simulation is a model for describing different patterns of behavior of a target system (Humphreys 2004)

Each of these viewpoints result in a set of different philosophical assumptions related to the supposed epistemological significance of simulations. One example of this concerns the question of the relation between "traditional" experiments and simulations.

The focus of these more philosophical works is thus on how simulation challenges (or not) more classical issues within the philosophy of science and so gives precedence to science rather than to technology. Moreover, these works often lack an *in-depth* historical perspective and assume

3 This is to some extent explained by the fact that the main community of historians of computing identifies itself as historians of technology rather than of science and so not too much attention has been given to actual scientific practices in that community. See (Hashagen 2013) for an overview and further bibliographical details and (De Mol and Bullynck, forthcoming) for a critical reflection on the difficult relation between historians of computing and history of science (and, more particularly, mathematics).

4 I am indebted to Juan Duran for having shared with me an earlier draft of this paper.

a kind of stability of concepts over time (Duran forthcoming).[5] One notable example of this is the fact that many philosophical discussions on simulation focus on the philosophical novelty of simulation (or lack thereof) but at the same time ignore the existing historical work which attacks the singularity idea of the computer and which emphasizes also the continuities with older practices in order to isolate more clearly what really changed in the history of science in the 1940s and 1950s due to the computer (and which lay the basis for the later work).

The aim of this paper is to revisit the so-called roots of computer simulation in ENIAC in order to contribute to such historical perspective. I will do this by developing two related but distinct claims. First of all, I will show how, around ENIAC, one can find a diversity of practices which one tends to ignore if one focuses only on the ENIAC as the originator of Monte Carlo simulation. This broadens the outlook on "simulation" in those pioneering years which thus has different forms not just chronologically but also diachronologically. Secondly, I will develop the claim that around ENIAC there is no simulation,[6] both in a historical as well as in a methodological sense. From the historical perspective, it should be pointed out that, as far as I know, the notion of simulation is, within the ENIAC context, not a standard term to use.[7] Secondly, and more important, the idea that there is no simulation is also taken as a methodological approach in this paper: instead of focusing on what is being simulated (weather; neutron behavior; bombing behavior; etc) the approach here is to look at "simulation" from the perspective of what it is, as structured by the interrelations between three fundamental levels of the practice:

(1) the mathematics used and developed (human)

(2) the (logical organization of the) program (human-machine)

(3) the physicality of the machine (machine)

In order to develop these claims, this paper is structured in two main sections. In Sec. 2, I will introduce the ENIAC and engage with some of the reasons to construct the machine. This will allow me to highlight some of the continuities and discontinuities with previous calculatory practices. In Sec. 3 then I will engage with the work of three different ENIAC "users": Derrick H. Lehmer (Sec. 3.1), Haskell B. Curry (Sec. 3.2) and John von Neumann (Sec. 3.3).[8] In a discussion

5 An important exception here is the work by Franck Varenne, see for instance his (Varenne 2007; Varenne 2009). This work is rooted in a detailed case-study which focuses on the use of modeling and simulation in the context of plant biology. Based on that work, Varenne gives a kind of minimal characterization of a computer simulation, more particularly, a computer simulation "*is minimally characterized by a general strategy of symbolization taking the form of at least one step by step treatment*." (Varenne 2009, p. 11) Based on that more pragmatic understanding, one can diversify between different types of computer simulation which depend on their particular epistemological function at a given time.

6 This statement, *Es gibt keine (Computer) Simulation* is very much inspired by Friedrich Kittler's *Es gibt keine Software* (Kittler 1993)

7 This is confirmed by discussions I had on this with Thomas Haigh, Mark Priestley and Maarten Bullynck.

8 These study-cases are very much rooted in previous work I did on ENIAC. See: (Bullynck and De Mol 2010; De Mol, Carlé, Bullynck 2015)

section I will return to the main claims of this paper and conclude with a critical viewpoint on “simulation” from the contemporary perspective.

2. Introducing ENIAC - historical setting

The ENIAC computer is probably the most famous and, at the same time, controversial computer that was ever built. It is often presented as (one of) the first computers[9] but it also played a central role in the important Honeywell, Inc. vs. Sperry Rand Corporation case which invalidated the ENIAC patent that was filed by Eckert and Mauchly, the two main engineers involved with the design and construction of the machine.

The ENIAC machine and other contemporary machines can be contextualized in a broader history of mechanization of human calculation in applied mathematics and science in general.[10] More particularly, it fits into a history of calculatory practices for table making (for instance, the use of difference engines to mechanize the method of finite differences) and table processing (for instance, the use of punched cards to mechanize the processing of census data) (Campbell-Kelly et al 2003; De Mol and M.-J. Durand-Richard forthcoming). The fact that the main journal for reporting on advances in high-speed computing in the late 1940s and early 1950s was the journal *Mathematical tables and other aids to Computation*, which was founded only in 1943, is one indication of the strong ties between table making and the development of the first computers.[11]

Within that context, one had to deal with the issue that human computation was laborious, time-consuming and very error-prone.[12] For instance, Charles Babbage, whose work on and design of the difference and analytical engines is often seen as a precursor of the modern computer, supposedly and famously exclaimed: `*I wish to God these calculations had been executed by steam’* when he and Herschel were checking a manuscript of calculations for astronomical tables and they realized the rather large number of errors (Swade 2003:158). It was the increased need for such tables combined with the growing realization of these issues that results in the development of more “efficient” methods of calculation. One development concerns “deskilling” methods (Swade 2003: 150) which are introduced to simplify the calculatory process in such a way that the calculations can be done by people who are of a lower level of education and a lower social class.[13] The other is the development of mechanical aids which include digital machines (for instance, difference engines or Aiken’s Mark I) and so-called “analog” devices[14] (for instance, the differential analyzer). The latter

9 See (Haigh, Priestley and Rope 2016) for a recent discussion.
10 See for instance (Grier 2005)
11 In fact, when the *Eastern Association for Computing Machinery*, which became the well-known *Association for Computing Machinery*, was founded *MTAC* was the preferred journal to publish in until *ACM* founded its own journal.
12 Of course, one can say the same of machine computations but scaled by several factors.
13 For instance, de Prony supposedly hired unemployed hairdressers in his “computation factory” which was set-up for the computation of logarithmic and trigonometric table. See (Campbell-Kelly 2003) for more details.

machines were used to solve differential equations and remained for quite a long time after the introduction of the digital high-speed computer, a preferred tool of the engineers.[15]

It was then mostly the issues of speed and error which resulted in the U.S. Army accepting Mauchly's proposal to construct an electronic high-speed machine. In the 1930s there was a growing need for ballistic tables used to aim fire at an enemy target. These tables involved complex calculations because of the large number of factors that affect the trajectory of a missile like, for instance, wind velocity, the shell's weight, diameter and shape and even the rotation of the earth for long-range missiles (Polachek 1997). It was the Ballistics Research Lab (BRL) at Aberdeen proving ground, Maryland which had, as one of its main tasks, the computation of such ballistic tables for the U.S. army. Aberdeen Proving Ground was the first U.S. Proving Ground and constituted just a few months after the U.S. had entered World War I. It was here that weapons were being designed and tested and so it played a major role in bringing closer mathematics and military applications.

In order to compute firing tables, they relied on both hand calculations aided by desk calculators and computations from the differential analyzers at BRL and the Moore school,[16] depending on the type of trajectory to be computed. However, each of these methods had its shortcomings. With the hand calculation method, it took about two 8-hour days to do one trajectory; the differential analyzer was much faster in that respect, taking about 15 to 30 minutes for one trajectory (Polachek 1997), but the solutions were not accurate and needed to be `smoothed' by additional hand calculations. As Eckert recounts (Eckert 1977: 7):

> "You finally get errors in the final result of the order of 1 percent or worse. The ballistics work that we were doing required errors of something like 10 times better than that. So what was being done at the time was that things were being run on the analyzer and then they were being smoothed by hand calculations......being calculated by some interpretive schemes to try to improve the accuracy required now."

Moreover, it took on the average one day for setting up the DA to change from one type of trajectory to another (Polachek 1997, Grier 2005). Given these circumstances, the production speed

---

14 For a discussion of the history of the use of the pair digital-analog see (Kline forthcoming). As is shown there, not everyone agreed with this terminology. Hartree for instance used the terms mathematical instruments and calculating machines to make the difference between machines that measure and machines which count and Stibitz was more convinced of the pair continuous and digital. Most of the discussions on this are about the use of the word `analog'. Indeed, one can argue for instance that also a "digital" machine can function as an analogue of some digital process for instance.

15 See (Kline forthcoming). Hashagen made a similar point for the German case in a recent talk at a research séminaire in Lille (8 march 2017) titled *Analog computing as a failed modernization program in Germany 1930-1960.*

16 There were at the time three differential analyzers in the U.S. The original DA from Vannevar Bush at the MIT and two copies, one at the Moore school of Penn University and one at BRL. The latter two were built at the Moore school but financed by the army under the agreement that researchers at BRL could use the DA at the Moore school at times of war (Grier 2005).

of the ballistic tables could not catch up with the need for new tables which only increased with the start of the second world war and so Mauchly's proposal to the Army to construct a high-speed computer to compute firing tables was perfectly timed (Polachek 1997: 25):

> "In spite of the extensive arrangements the laboratory made [...] the backlog continued to grow. At one point, more than 100 female students were engaged to carry out firing table calculations. It was to relieve this bottleneck that John W. Mauchly and J. Presper Eckert [...] proposed the construction of the ENIAC"

And so, in 1943, the construction of ENIAC was started at the Moore school, Penn University. It would take until 1946, after the war, that the machine would be completed. Even though it was too late to serve its initial purpose, it was realized that it could also be used for a host of other applications, including the famous Monte Carlo calculations used for the design of nuclear weapons. According to the list that was made by Barkley Fritz and which summarizes the different sorts of computation that were done on ENIAC, it was used for over hundred different problems,[17] going from the production of number-theoretical tables to the use in weather prediction problems.

The machine had, in a sense two "lives" (Neukom 2006). In a first stage, the machine was highly modular and parallel. In order to `program' it, one had to rely on a method of rewiring the machine by reconnecting the different modules and set-up a large number of switches (for instance, to store numbers or to indicate to a given addition unit whether it should sent out a number or its complement at a given step in the program).[18] By consequence, programming the ENIAC in its original configuration (Fritz 1994, p. 31):

> "[…] can best be described as analogous to the design and development of a special-purpose computer out of ENIAC component parts for each new application […] Anyone now doing research in parallel computing might take a look at ENIAC during this first time period, for indeed ENIAC was a parallel computer with all of the problems and opportunities this entails."

Preparing and setting-up a program on the original ENIAC was a very time-consuming job. Moreover, the "length" and complexity of a program, was very much determined by certain physical boundaries like the number of units available or the number of program cables. Because of these issues, it was decided that the machine should be permanently rewired once it was moved from the Moore school to BRL, so that it would become possible to "code" the program by using symbolic instructions rather than "wire" it. In that set-up the machine became a kind of stored-program machine applying concepts from von Neumann's report on the EDVAC, viz. the design which is by many considered to be the blueprint of the modern computer.[19] [20]

---

17 See for instance (Fritz 1994) to get an idea of the diversity of problems that were prepared for the machine.
18 See (Bullynck and De Mol) to understand in more detail how the ENIAC worked in its original set-up.

So why was this machine so special as compared to other calculatory devices that were being used at the time. There are three properties of the ENIAC which are considered to be basic and where each feature very much depends on the previous one:

1. It was a discrete machine

2. It was capable of so-called conditional branching which is considered as the key feature required to construct a general-purpose machine besides the capability of the four basic arithmetic operations (addition, subtraction, multiplication, division)

3. Possibility of coding the machine – this is only true of the rewired ENIAC but see in this respect footnote 19.

4. It had electronic high speed.

While properties 1-3 can already be found in other machines, most notably Aiken's Mark 1 and the Bell model 5 machine, it is the combination of 1 to 3 with 4 which is new and so, from this perspective, the most fundamental innovation of ENIAC was its speed. In fact, one can argue that it is the addition of this last feature to the other three which requires, makes possible and gives rise to a rethinking of computational methods including those for "controlling" those methods, viz. the programming. For instance, it is this property which made sensible the idea of internal storage of instructions inside of the machine (Alt 1972, p. 11).

> "I think the thing that we learned with this high speed was that ...you had to have a way to program ahead of time [...] On slow computers you can add manually, you can request manually the performance of each arithmetic operation. But a fast computer is useless unless you have some way to program it. ... The idea of a stored program [...] couldn't have come up before we had the ENIAC because it would have been useless."

In Sec. 3 further indications are given of the significance of ENIAC's high speed in combination with the other three features indicated here. Thus, if one accepts this analysis, the accounts of what is the supposed impact of the modern computer on science, should be able to give a good explanation of how a \`mere' quantitative change has resulted in such apparently fundamental changes in science and its practice.

3. Three people – three practices

This section, which is the main section of this paper, considers the works and viewpoints of three people who were involved with ENIAC: Derrick H. Lehmer, Haskell B. Curry and John von Neumann. Curry and Lehmer were selected here because they were two of the three members of the

---

19 See (Haigh, Priestley and Rope 2016) for a historical reconstruction of the conversion process and an analysis of the capabilities of the rewired ENIAC in relation to the EDVAC design and the programming "method" that was designed for it by von Neumann and Goldstine.

20 Note that the converted ENIAC is, from a certain perspective, just another set-up of the original machine.

so-called computations committee that was assembled by the U.S. Army in order to test, amongst others, the newly built ENIAC. Moreover, they allow to counterbalance a bit the "legend" of John von Neumann who is probably the most well-known mathematician who was involved with ENIAC.

The parallel discussion of aspects of each of their works and viewpoints permits to show the diversity of methods that were developed around this machine, thus contextualizing the Monte Carlo "simulations".

3.1. Derrick H. Lehmer – number theory

3.1.1. Background and involvement with ENIAC

Derrick H. Lehmer was first of all a number theorist. He was very much shaped by his father who was also a number theorist and convinced "*that mathematics, and especially number theory, is an experimental science*" (Lehmer 1969, p. 3). Lehmer was very much aware of the possibilities of mechanical aids to this kind of number theory as becomes clear from his involvement in the journal *Mathematical Tables and other aids to Computation* and his construction of several "prime sieves" in the 1920s and 30s, including a photoelectronic sieve and one built from bicycle wheels.[21]

During World War II, Lehmer got involved with the war effort by contributing to research of the Applied Mathematics Panel (AMP). It was established by the end of 1942, when the National Defense Research Committee (founded in 1940) was reorganized going from five divisions to nineteen. It was directed by Warren Weaver. Its purpose was *`to bring mathematicians as a group more effectively into the work being carried on by scientists in support of the nation's war effort.'* (Bush, Conant, Weaver 1946, p. vii). The AMP had contracts with different universities who could work on specific problem classes. The University of California where Lehmer was working at the time, was responsible for (Bush, Conant, Weaver 1946: 97):

> "Statistical analysis applied to bombing research concerned with problems of land mine clearance, the theory of pattern bombing and the bombing of maneuvering ships, and the theory of bomb damage."

One set of problems that was studied by the group in California was related to *pattern bombing,* which concerns *`the almost simultaneous release of all the bombs carried by a formation of aircraft, thus giving rise to a pattern of bombs affected, as a unit, by an aiming error.'* (Bush, Conant, Weaver 1946, p. 46) and it was with this problem that Lehmer got involved. More particularly, there is, first of all, the report (Eudey 1944), *Cooperative study in area bombing*, and, secondly, and more interesting here, the report (Lehmer 1945) describing a photo-electric

21 See (Bullynck 2015) for a discussion of Lehmer's different sieves.

instrument for mechanizing a specific method `t*o estimate the probability of at least one hit or, alternatively, the expected proportion of hits, in formation attacks on irregular target areas.'* Interestingly, and as will be argued in more detail elsewhere,[22] this method, which is called a *model experiment*, can be understood as a variant of the Monte Carlo method before it had been used and developed in the ENIAC context and is very close to a notion of simulation as a means to solve problems in a non-analytical manner. More particularly, a model experiment is *`used as a means to solve certain bombing problems which would proceed tediously if approached by numerical integration'* (Bush, Conant, Weaver 1946b, p. 56) One example (from 1944) from the review report (Bush, Conant, Weaver 1946) in which a model experiment was applied was the problem (Bush, Conant, Weaver 1946, p. 48)*:*

> "to determine the number of attacks […] needed to give a probability of at least *P* that at least the proportion F of the target would be covered at least n times"

The method was the use of a model experiment:

> "in which a series of synthetic random-bombing operations were performed, with enough replications to permit the estimation of probability levels from order statistics. The data, so accumulated, was then used as the basis for an empirical function whose general properties were suggested by theoretical considerations."

It is exactly in this kind of context that one also finds reference to a notion of simulation. So, for instance, in the introduction to Chapter 7 *Statistical studies in mine clearance,* of the same summary report one reads (Bush, Conant, Weaver 1946b, p. 79)*:*

> Most of the work in the second and third studies was done by experimental statistical methods in which model experiments *simulating the conditions of the problem* [m.i.] were repeated a number of times. The theory underlying the two studies can be formulated analytically in terms of appropriate mathematical formulas but the computation that would have been involved in the mathematical approach would have been prohibitive. The experimental methods developed are fairly simple but quite effective and the routine, once it is set up, can be made to operate at a clerical level. *There are undoubtedly many other statistical problems of this type in military research which can be more effectively handled for practical purposes by experimental methods than by analytical methods.*[m.i.]

Moreover, it seems that this notion of `simulation' refers directly to the random events used in the model experiments as is clear here (Bush, Conant, Weaver 1946b, p. 79):

> A second statistical study dealt with an investigation of the extent of clearance of mines to be expected by using against beach minefields 120-rocket barrages launched by a device

22 I am preparing a paper together with Maarten Bullynck with the provisional title: *Monte Carlo on the beach: model experiments and throwdown studies before ENIAC* in which we consider variants of the Monte Carlo method which clearly already had a certain tradition before their use in the ENIAC context.

known as the WOOFUS. This study was carried out by means of a miniature random number experiment, in which the radius of clearance of a single rocket and the errors involved in delivering the 120 rockets in a barrage were simulated.

Fig. 1 gives a schematic representation of Lehmer's instrument called the photoelectric analyzer and, most probably, inspired by Lehmer's photo-electronic prime sieve.[23]

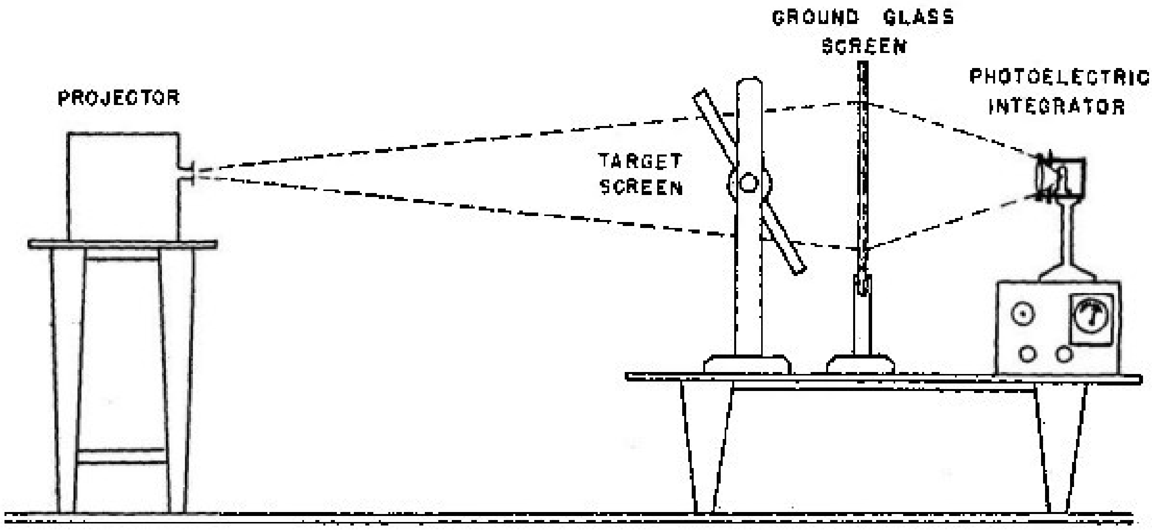

Fig. 1: Graphical representation of Lehmer's photo-electronic bombing analyser. From (Bush, Conant, Weaver 1946).

Its basic principle was to (repetitively) project a synthetic bomb pattern on a ground glass after passing through a diaphragm stop cut out in the form of the target (the so-called "target screen" in Fig. 1) so that the ground glass screen would be illuminated only by that part of the bomb pattern which intersected the target. The light from the screen was focused on a photoelectric cell which (Bush, Conant, Weaver 1946, p. 57):

> "was instrumented so as (1) to add the effect of successive images of bomb patterns, or (2) to count the cases which were not blank. A movie projector and a film with 1,500 frames were used. Each frame carried a picture of the bomb pattern with its center displayed to represent a random deviate from a Gaussian distribution."

Apparently, after the war, plans were made to construct a number of these instruments at Wright Field but I could not verify in how far these devices were actually used.

In 1945 then a computations committee was assembled at the Ballistics Research Lab at Aberdeen intended to prepare for "*utilizing the [ENIAC] machine after its completion*" (Alt 1972: 693). Given that Lehmer was already well-versed in the development and use of calculatory devices and had done some war work as a member of the lab in California, his being a member of that committee is less surprising as one think at first.

23 See (Bullynck 2015) for more details about that sieve.

3.1.2. Developing mathematics on the ENIAC: a number-theoretical problem

So what did Lehmer do to test ENIAC? Most important from that perspective is the number-theoretical computation that he prepared and set-up over a labour day weekend in 1946 with his wife Emma Lehmer who was also a number theorist.[24] This concerned the computation of exponents e of 2 *mod p*, viz. the smallest value of e such that $2^e \equiv 1$ *mod* p. It was a known fact that Fermat's little theorem could be used as a primality test. If for a given number b, $2^b \equiv 2$ *mod b* than b is with high probability a prime number. Unfortunately, an infinite set of exceptions to this primality test exists. A table of exponents then can be used to compute such exceptions and the ENIAC computation was used to correct and extend existing tables of exponents.

Given the military context in which ENIAC was built and used, one could consider this problem as quite obscure or irrelevant. However, as a test problem it was quite important and this for two reasons. First of all, and as was pointed out by Alt (Alt 1972: 694):

> “The running time of the problem occupied almost the entire weekend, around the clock, without a single interruption or malfunction. *It was the most stringent performance test applied up to that time, and would be an impressive one even today.*[m.i.]”

Secondly, given that it was the first number-theoretical computation that was ever ran on an electronic machine, it was important in that it showed that machines like ENIAC might also be useful for scientific purposes that were unrelated to the war effort and so it could be used as an example to convince scientists of the usefulness of high-speed computation for their work (Alt 2006: 40):

> “I think what’s particularly interesting about the number theory problem they ran was that this was a difficult enough problem that it attracted the attention of some mathematicians who could say, *yes, an electronic computer could actually do an interesting problem in number theory* [m.i.] – something as sophisticated in number theory – and produce useful results. There were many people who speculated about this – von Neumann among them – but to actually do it, to demonstrate it, was, I think, important to the post-war reputation of electronic computers among mathematicians.”

Clearly, while the problem itself already had a tradition within number theory, the introduction of a high-speed and parallel machine affected the methods for tackling it. First of all, and in connection to Lehmer’s previous work, the machine allowed the implementation of a truly parallel prime sieve. Secondly, one of the main subroutines that were set-up on the machine and which was called the *exponent routine* was quite different from the human methods one would normally use. Indeed, and as Lehmer explains, the ENIAC “*was instructed to take an ``idiot'' approach*" (Lehmer 1974: 5). To start, the machine needed a table of prime numbers. Of course, in a context where one does not have

24 A detailed study of that work can be found in (Bullynck and De Mol 2010).

the high-speed, the obvious thing to do is to provide that table to the machine during the computation by means of punched cards. However, since this is a mechanical process, this would significantly slow down the computational process. Hence, it was decided not to use an existing prime table but to let the machine compute its own next value of *p* as it was needed by using the mentioned prime sieve. The next step was to calculate the powers of 2 reduced modulo *p,* with *p* prime. The "idiot aproach" taken by ENIAC resulted in a routine which required `*only one addition, subtraction, and discrimination at a time cost, practically independent of p, of about 2 seconds per prime. This is less time than it takes to copy down the value of p and in those days this was sensational*.' (Lehmer 1974: 5). Thus, with the introduction of high-speed computation, it was realized that the usual human and slow methods needed to be replaced by other methods which might be less ingenuous and more brute force but also more efficient. Hartree, another mathematician, who was already well-versed in using machines to assist in computation when he became involved with ENIAC, phrased this as follows (Hartree 1949):

> "[I]n programming a problem for the machine, it is necessary to take a "machine's-eye view" of the operating instructions, that is to look at them from the point of view of the machine which can only follow them literally, without introducing anything not expressed explicitly by them, and try to foresee all the unexpected things that might occur in the course of the calculation, and to provide the machine with the means of identifying each one and with appropriate operating instructions in each case. And this is not so easy as it sounds; it is quite difficult to put oneself in the position of doing without any of the hints which intelligence and experience would suggest to a human computer in such situations."

It was also this approach which was used in the computation of another problem that was suggested by Lehmer to George Reitwiesner who was at Aberdeen Proving Ground and which became known as "slow Moses". This was the computation of the so-called Fermat quotient. The program was special because it was probably the first instance of *"an interruped idle time modus operandi"* (Lehmer 1974: 5). Apparently, Reitwiesner used it to prove to the engineers that ENIAC was also able to run for longer periods of time by running this problem every night for a certain period of time when it would otherwise stand idle (Homé Mc Allister Reitwiesner in (Bergin 2000: 44)). Regretfully, *slow Moses* had deserved its name because it was really slow due to the rewiring of ENIAC into a slower serial machine (Lehmer 1974: 5).[25]

Another interesting aspect of the exponent computation is that it required the full machine and even more. As is recounted by Jean Bartik, one of the six female operators of the original ENIAC (Bartik 1973):

25 Indeed, from Lehmer's perspective changing ENIAC from a parallel machine to a serial one was not a good choice and so, to him, von Neumann "spoiled" the ENIAC (Lehmer 1980).

> "Lehmer's little problems, they were always too big for it. So consequently, you always had to be changing it or to think of something new and innovative in order to get a problem or ways that you could break the problem down into smaller portions."

Thus, one could say that in the case of the exponent problem, the machine "implements" or "is" the method and so both the method and machine are reciprocally structured and shaped.[26]

It is not completely clear what other things Lehmer did in the ENIAC context though it is quite certain that he did assist in the preparation and actual set-up of a number of other problems that were ran on ENIAC as is clear, for instance, from the acknowledgements in (Hartree 1946) and (Grubbs 1950). Thus, at least in Lehmer's case, there is no "radical difference" between practices of number theory, machine-building and problems of applied mathematics. In fact, it was the need for mechanization in both number theory and applied mathematics which made it quite straightforward to make the switch from one to the other practice *via* the machine-building. This fits into a longer historical tradition where mechanization, computation and (applied) mathematics go hand-in-hand (See Sec. 2).

3.1.3. A number-theorist engaging with ENIAC – reflections

But while Lehmer's use of ENIAC is, from a certain perspective, a continuation of an existing tradition, it is also a culmination of that tradition: Lehmer was very much impressed by the potential of high-speed digital computation for his own field and after his meeting with the ENIAC machine he would write and talk on several occasions about the usefulness and impact of electronic high-speed computers on science and, more specifically, mathematics.

Lehmer clearly had specific views on number theory and mathematics at large. Indeed, as explained in Sec. 3.1.1., he had a view on number theory as an experimental science which he contrasted with the more traditional and popular viewpoint, or, in Lehmer's words `school of thought' (Lehmer 1966: 745):

> "The most popular school now-a-days favors the extension of existing methods of proof to more general situations. This procedure tends to weaken hypothesis rather than to strengthen conclusions. It favors the proliferation of existence theorems and is psychologically comforting in that one is less likely to run across theorems one cannot prove. Under this regime mathematics would become an expanding universe of generality and abstraction, spreading out over a multi-dimensional featureless landscape in which every stone becomes a nugget by definition. Fortunately, there is a second school of thought. This school favors

26 This is less the case today where one no longer has access to the machine in such direct manner though one could argue that it is now the "program" and so, by extension, the software "implementing" the method which is reciprocally shaped and structured by its applications and uses.

exploration [m.i.] as a means of discovery. [B]y more or less elaborate expeditions into the dark mathematical world one sometimes glimpses outlines of what appear to be mountains and one tries to beat a new path in their direction. [N]ew methods, not old ones are needed, but are wanting. Besides the frequent lack of success, the exploration procedure has other difficulties. One of these is distraction. One can find a small world of its own under every overturned stone."

Clearly, Lehmer's understanding of "experiment" cannot simply be equated to "experiments" as in, say, physics.[27] For one thing, the "reality" of a physical experiment is quite different from the "reality" of the mathematical experiment, especially the number-theoretical one, and so the idea that the computer has come to "*stand […] for nature itself*" (Galison 2011: 157) or constituted "*an alternative reality*" can and should be understood differently in the case of number theory. Indeed, here the computer, as a digital device which counts, has come to "stand" for the `universe' of numbers.

In Lehmer's view then, the computer is just another tool which requires, first of all, a rethinking of existing methods and, secondly, makes accessible a new range of problems which he calls `discrete-variable problems'.[28] For instance, in a paper written for a book volume titled *Modern mathematics for the engineer,*[29] the discrete-variable or digital machines are contrasted with analog machines, where these analog machines are assumed to be more familiar to the engineer *and* the mathematician (Lehmer 1956: 481).[30] The main focus of that paper is to show to the engineers for both types of machines what kind of problems they can be used for and how one needs to change methods when switching from analog to digital devices. Indeed, as Lehmer points out (Lehmer 1956: 485-486):

> From the point of view of the discrete-variable device, things need to be counted rather than measured; mathematics is not geometry but arithmetic; the universe is quantized and this includes mathematics. Integrals are but sums, and derivatives are but difference quotients; functions are discontinuous everywhere; limits, infinities and infinitesimals do not really exist […] Thus […] we seem to go back to Pythagoras. [...] The methodological step-by-step reiteration […] is to be contrasted with the modus operandi of the analogue machine."

In other words, if the computer constitutes an alternate reality it is a discrete reality and not a continuous one which is a problem one needs to deal with as a physician but not as a number theorist.[31]

27 For a more detailed discussion of Lehmer's understanding of experiment, see for instance (De Mol 2015).
28 Besides Lehmer also Eckert used the notion of discrete-variable (Kline forthcoming).
29 I thank Arianna Borrelli for having pointed me at this source.
30 See in this relation also footnote 14.
31 Of course, Lehmer was not the only one nor the first to describe the difference between so-called `analog' and digital machines in terms of measuring vs. counting devices. See eg (Stibitz 1945; von Neumann 1958: 3 and 6)

Interestingly and in the same paper, Lehmer does use the word `simulation' but it is not used with reference to digital machines but with reference to analog devices. This is not surprising: the very idea behind calling the "continuous" machines analog machines was exactly because they were understood as `analoguous' to what they were supposed to `simulate' namely, continuous phenomena. In fact, the current OED definition of `simulation', viz.:

> "The technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably *analogous* [m.i.] situation or apparatus, esp. for the purpose of study or personnel training."

seems closer in spirit to a notion of analog machines than to a notion of a digital machine as used by many in the late 1940s and early 1950s. Indeed, from the physical and engineering perspective, the transition from continuous to discrete-variable machines, at least at the time, was in fact a decrease in the level of analogy between the model or system being "simulated" as compared to so-called analog or continuous machines which where still more directly linked to the continuous world of physics.

3.2. Curry – logic and the automation of firing table computations

3.1.1. Background and involvement with ENIAC

Haskell B. Curry is today mostly known as one of the founders of combinatorial logic and mathematical logic in the U.S. Only few are aware of the fact that he became also involved in the second World War during which time he worked not on problems of logic but on ballistic problems. It was this involvement which resulted in him becoming one of the members of the *Computations Committee* for testing the ENIAC machine. So how did a logician become interested in and involved with ballistics? The answer to this needs to be sought in Curry's biography.[32]

Curry was born on September 12, 1900 at Millis Massachusetts. When he left high school in 1916, he entered Harvard University with the intention to go into medicine. When the U.S. entered World War I however, he enlisted in the Army where he became a member of the Student Army Training Corps in October 1918. He believed that he could be more useful for the war as a mathematicians so decided to switch careers with the idea of going into artillery (Seldin 2005). Curry was thus part of a small but growing community of people in the U.S. who realized the potentials of mathematics for the military.[33] Perhaps this went via George D. Birkhoff. He was at Harvard at the time and one of those who realized already during World War I the potentials of mathematics for the military (Archibald et al 2014). It would also be Birkhoff who, besides mathematicians like Veblen, Rees and Moulton, played a leading role in bringing closer U.S. mathematics and the military.

32 Most of the biographical information on Curry in this section is from (Seldin 2005).
33 See (Archibald et al 2014) for a recent historical study of the effect of World War I on U.S. Mathematics.

When the war had ended, Curry left the Army but continued to study mathematics and got his A.B. degree in 1920. After that, he went to MIT to become an electrical engineer working half-time at the General Electric Company but then decided that he was more interested in science than engineering and so switched to physics in 1922, returning to Harvard. He got his A.M. in physics from Harvard in 1924 but again his interests had shifted. He wanted to return to mathematics and so moved back to studying mathematics at Harvard until 1927. It was during that time that he became more and more interested in logic and decided to work on that, against the advice of several faculty members at Harvard. At that time, mathematical logic wasn't very well-developed in the U.S. (Davis 1995) and so it certainly wasn't the best choice career-wise. In fact, Curry was supposed to write a dissertation on differential equations under Birkhoff. He became, instead, fascinated by the substitution operation for propositional logic as described in *Principia Mathematica* by Russel and Whitehead which he considered to be too complicated. His approach then was to analyze the substitution operation down to its simplest possible elements. The result of that was a set of operators which he called combinators.[34] This approach, to reduce something to its most elementary form, would become a characteristic of part of Curry's way of working and, as we will see, would be one of the main methods of his theory of program composition that he developed in the wake of his meeting with ENIAC.

When he brought this work to, amongst others, Birkhoff and Wiener (then at MIT), the reaction was a positive one and so he changed topics for his dissertation. However, since there was no one at Harvard who could supervise a dissertation on logic, he moved to Princeton and then to Göttingen where he finished his dissertation under the supervision of David Hilbert and Paul Bernays. Upon his return from Göttingen he became an assistant professor at Penn State University where he would stay until his retirement in 1966.

However, switching to logic, which was certainly not the most applied field of mathematics at the time, did not stop Curry from investing into the military as a mathematician. And so when the second world war started he became a member of the the joint War Preparedness Committee of the *American Mathematical Society* and the *Mathematical Association of America*n and which was chaired by Marston Morse, a student of Birkhoff who had been in the Ambulance service and was sent to France at the end of the first World War. In a paper titled *Mathematical Teaching and National Defense,* which resulted from the committee's deliberations on `*all aspects of the relation between mathematics and defense*' (Curry 1942: 337), it becomes clear how strong Curry's convictions actually were (Curry 1942: 337):

34 Some time later, Curry found out that he was not the first to have developed this type of operations to analyze substitution. Indeed, Schönfinkel had already develop several of the combinators in his paper *Uber die Bausteine der mathematischen Logik* which was published in 1924 already. By the time however Curry found out, Schönfinkel was in a mental hospital. The story of Schönfinkel is very sad: he died in poverty in Russia in 1942.

> Those who allege that algebra, for instance, has no practical use may be surprised to learn that *modern war is largely mathematical in character* [m.i.]. The firing of projectiles, the design of airplanes, the construction of secret codes, and countless other activities require large amounts of mathematics which is sometimes highly technical.

When the U.S. entered the War, Curry left university life to work and serve as a mathematician at Frankford Arsenal. This was one of six U.S. Army ammunition facilities. Each of these worked on one or more of the major phases of ordnance. The one in Frankford (Hoff 1943: 18):

> "makes studies on the manifacture of metal components for shells and bombs, and designs and develops methods for production of small arms ammunition."

Once a specific weapon was designed and a prototype produced, it was sent to Aberdeen Proving Ground where it was tested. It is not completely clear what Curry did during his time at Frankford. According to (Seldin 2012) he worked mostly `*on the mathematics of aiming a projectile at a moving target, the so-called fire control problem.'* He also published two papers while at Frankford Arsenal (Curry 1943; Curry 1944).[35] Neither of these papers contains new methods but are instead descriptions of existing mathematical techniques `*with emphasis on its practical aspects*' (Curry 1944: 259) in order to show their potential for the kind of problems Curry was confronted with at Frankford Arsenal. So, for instance, the first paper considers the so-called Heaviside operational calculus which was, apparently, `*no longer of any mathematical interest*' (Curry 1943: 366) and been `*completely replaced* [in engineering practices] *by the theory of Laplace or Fourier transform*.' (Curry 1943: 366). However, it could still be very useful for certain artillery problems, as Curry continues:

> "I know from the experience of the last six months that there are engineers who find the theory of integral transforms involves difficulties which they would prefer to avoid. Moreover there are serious engineering problems, viz., those having to do with discrete mechanisms or networks, for which the difficulties of the integral transforms appear to be irrelevant. There is, consequently, some interest in a theory of the Heaviside calculus of a more elementary character."

After his short stay at Frankford Arsenal, Curry became a member of the Applied Physics Laboratory at John Hopkins where he stayed until May 1945 to move to BRL at Aberdeen proving Ground where he stayed until September, 1946 and made it to chief of the Theory section of the Computing Laboratory and then chief of the Computing Laboratory. It was, of course, during this time that he became involved with ENIAC as a member of the computations committee. Based on this experience he would become consultant in computing methods to the U.S. Naval Ordnance from June 1, 1948 till June 30, 1949 (Seldin 2005).

35 One of these papers was published in the the newly founded *Quarterly for applied mathematics.* That journal was founded in 1943 and indicates the increased interest in the U.S. for applied mathematics.

3.2.2. Developing mathematics on the ENIAC: interpolation problems

While at Aberdeen, Curry worked *mostly* on problems related to computing firing tables by the ENIAC. I have found four problems Curry has worked on while at Aberdeen and in direct relation to ENIAC:

1) numerical method for "smoothing" drag functions

2) inverse interpolation problems

3) fourth order interpolation

4) the computation of the first 2,000 digits of *e*

I will focus here only on 1) and 2). For a short discussion of 4), see (Seldin 2012).[36] A description of the work on 3) can be found in the report (Curry and Lotkin 1946).

Numerical method for "smoothing" drag functions Curry's work on problem 1) is in fact a modification of the spline interpolation method which was introduced by Isaac Jacob Schoenberg also in the ENIAC context. Schoenberg was a Jewish mathematician who had moved to the U.S. in the early 1930s and also had a position at the University of Penssylvania by the time the second World War started (Schoenberg 1988). In August 1943 he also joined the BRL at Aberdeen Proving grounds. It was Leo Zippin, yet another mathematician who had joined into the war effort and served as a Corporal at BRL from 1942 till 1945, who arranged for Schoenberg to go to BRL for the duration of the war where he was given a very specific task (Schoenberg 1988):

> The morning in August 1943 of my reporting for duty, Major A. A. Bennett, of Brown University, then Chief of the Computing Branch of the BRL, told me what my particular problem was to be: 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

36 In relation to 3) it is interesting to point out that the first thoughts on this by Curry date back to before the ENIAC was finished. More specifically, there is one note from Curry in the Curry papers dated July 15, 1945 (Seldin 2012). The actual "written" program was finished by January 1946, still before the official unveiling of ENIAC. It is unclear whether this specific problem was ever ran on ENIAC but apparently Curry told Seldin that no one else at ENIAC was convinced that it could be of any interest at the time. However, some years later, von Neumann became interested in computing the first 2,000 digits of both *e* and *pi* most probably because of the interest in finding good methods for generating random digits (See Sec. 3.3.2.). The (then rewired) ENIAC was effectively used to compute the digits of both numbers. See (De Mol 2008) for more details.

> 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* [m.i.] To do this was my problem.

In other words, Schoenberg was asked to develop a numerical method which would be more adapted to the particularities of a high-speed digital computer already in 1943 when it had only just been decided that the Army would finance the construction of that machine, viz. a method which takes advantage of the digital and high-speed of the machine and so makes extensive use of iteration which, as a side-effect, accumulates round-off errors because of the "roughness" of the drag functions.[37] The method which was introduced by Schoenberg is known as (an instance of) *spline interpolation*. As Schoenberg explains (Schoenberg 1946), this method is used when the more common interpolation methods (using polynomials) cannot give enough accuracy. This is especially the case when the function *F(x)* is an approximation based on a set of empirical data y_n as in the case of the drag functions (instead of a known analytical function *F(x)*). The basic idea is to approximate a function by piecewise polynomial functions, viz. instead of having one polynomial, the approximation is done by polynomial pieces of a certain degree *n* which join at certain points known as knots. Schoenberg's original method used equidistant knots and it was then Curry `*who recognized the possibility of defining splines with arbitrary non-equidistant knots*' (Schoenberg 1988: 5) This work was published as (Curry and Schoenberg 1947; Curry Schoenberg 1966).

Inverse interpolation problems[38] The second problem from the ENIAC context on which Curry worked is the so-called problem of inverse interpolation (Curry and Wyatt 1946)[39]:

> "The problem of inverse interpolation may be stated as follows. Suppose we have a table giving values of a function x(t) [...] for equally spaced values of the argument t. It is required to tabulate t for equally spaced values of x. This problem is important in the calculation of firing tables. Suppose the trajectory calculations have given us the coordinates (x,y) of the projectile as functions of t (time) and φ (angle of departure). For the tables we want t and φ as functions of x and y; indeed we wish to determine φ so as to hit a target whose position (x,y) is known, and t is needed for the fuze setting or other purposes."

---

37 Also von Neumann would later become interested in the study of round-off errors and the development of numerical methods for controlling them but it is clear that the need for such methods was realized very soon and before von Neumann for involved with ENIAC. See Sec. 3.3.2.

38 See (De Mol, Carlé, Bullynck 2015) for a detailed study of Curry's work on the inverse interpolation problem and how it affected his later work on the composition of programs. Parts of this paragraph overlap with conclusions from that paper.

39 Willa Wyatt was one of several female mathematicians that was hired by the BRL and then sent to the Moore school to work on the calculation of ballistic trajectories. She was one of the supervisors of the differential analyzer and the computer sections at the Moore school and so was well-versed both in ballistic calculations and computing machinery (Fritz 1996)

In other words, the problem of inverse interpolation concerns the computation of initial settings of a bombing device such as the fuse settings or the angle of departure. As was the case with the smoothing problem, also here the numerical methods are very much adapted or shaped by the limitations and possibilities of the high-speed ENIAC machine. To start with, the numerical method used is based on iteration `*which is eminently suitable for ENIAC*' (Curry and Wyatt: 6). But the choice for an iterative method for the main interpolation is not just rooted in the high-speed of ENIAC but also in the possibility to reuse certain programs `*independent of the choice of the [interpolation] formula for f(u)*' (Curry and Wyatt 1946: 6) where f(u) is the interpolatory approximation of the function x(t) mentioned above. Indeed, the report by Curry and Wyatt is intended to provide a *general* framework for problems of inverse interpolation and so (Curry and Wyatt 1946: 6):

> "A basic scheme of programming is set up in detail in such a way that it can be readily modified to suit circumstances. Some modifications are also programmed in somewhat less details than the basic scheme, and general principles regarding modifications are discussed. The special requirements of firing table calculations […] are not gone into here."

In other words, the methods of tackling the problem of inverse interpolation is very much determined not just by the high-speed of the computation but its combination with ENIAC's `programmability' (See Sec. 2). In fact, we can go a step further and say that while the report by Curry and Wyatt is clearly focused on the inverse interpolation problem, this is studied not for its own sake but `*with reference to the programming on the ENIAC as a problem in its own right*'.

As explained in Sec. 2, for the original ENIAC machine, setting up a problem was quite laborious and so many of the choices and reflections made in the Curry-Wyatt report are motivated by the need for simplifying the programming process. One clear example of that is the choice for a numerical method which is not the most efficient in terms of convergence -- which was less of a problem given the high-speed of the machine – but results in a simpler program set-up, or, to put it in the words of Curry and Wyatt: `*For the ENIAC [...] extremely rapid convergence is not necessary. [...] A far more important consideration than speed of convergence is simplicity of programming.*' (Curry and Wyatt 1946: 14). Perhaps more interesting from the contemporary perspective is the development of a more systematic and structural approach to programming in the report, providing a hierarchical structure to programs which differentiates between:[40]

1) program elements
2) program sequences or stages
3) processes
4) program

40 It seems that this structurization is due to Curry. Indeed, in his notes on the *e* calculation, which predate the Curry-Wyatt report, he already identifies this kind of hierarchical structure.

The more central elements here are the `stages' or `program sequences' of a process:

> "The stages can be programmed as independent units, with a uniform notation as to program lines, and then put together"

Indeed, it is the structurization of processes into stages which allows for the modifications needed to do different types of inverse interpolation simply by reordering or reusing the stages as well as to make possible the use of a particular stage at different places in a program. The latter method is today known as a *closed subroutine* and is usually considered to be a fundamental feature only of stored-program machines.

As is clear, even though Curry was involved with ENIAC and BRL only for a short time, he was quite able to bring together the needs and limitations of the ENIAC machine with his experience in applied mathematics and his viewpoint as a logician who seeks to simplify complicated operations by analyzing them into simpler elements. This bringing together of these three aspects was driven not so much by a desire to solve one specific calculatory problem (e.g. one particular class of firing tables), but instead to develop more general numerical and programming methods that fit the machine to give a higher degree of efficiency (Curry and Wyatt 1946):

> "[The] basic scheme was not designed specifically for a particular problem, but as a basis from which modifications could be made for various such problems."

Curry would pursue this path by developing a so-called *theory of program composition* which aimed at automating the process of subroutining and so can and was understood, at least by some, as an anticipation of much of the work that was done much later on compilers and higher-level programming in the later 1950s.

3.2.3. A logician engaging with ENIAC – reflections

What is logic and, more specifically, formal logic about? In his first published paper in logic, Curry summarizes this as follows (Curry 1928: 363):

> "the essential purpose of mathematical logic is the construction of an abstract (or strictly formalized) theory, such that when its fundamental notions are properly interpreted, there ensues an analysis of those universal principles in accordance with which valid thinking goes on. The term analysis here means that a certain rather complicated body of knowledge is exhibited as deriveable from a much simpler body assumed at the beginning. Evidently the simpler this initial knowledge, and the more explicitly and carefully it is set forth, the more profound and satisfactory is the analysis concerned."

In other words, formal logic is the bringing together of what Curry considers to be the main purpose of logic, viz. `t*he analysis and criticism of thought*' (Curry 1928: 363) with the formal methods of mathematics. Or, put differently, it is about modeling (valid) human thought through formalization.[41] Moreover, it is Curry's specific approach to perform such analysis through simplification, viz. (Curry 1928: 367-368):

> "The rules [of any abstract theory] form the port of entry of intelligence; and since nothing can be done without them, they represent the atoms of thought [. . .] into which the reasoning can be decomposed. It follows that in constructing such a theory it is not sufficient merely to reduce the postulates and primitive ideas to their lowest terms; it is even more important to so chose the rules that they involve [. . .] only the simplest actions of the human mind."

It is exactly this approach of simplification which he then applies to the `*highly complex*' rule of substitution resulting in Curry's theory of combinators (which is closely related to that of Schönfinkel, see footnote 32). It serves the purpose of understanding `*processes by means of which entities may be combined to get new entities*' and so we see that Curry's `model' for substitution is intended to capture (rather than simulate) a dynamical process. The model however has the potential of being a dynamical one (see p. 2) if we start to effectively "apply" the combinatory rules to `generate' a system of assertions.

This is confirmed by Curry's use of his work on combinators in his attack on the *problem of program composition* and which is a continuation of the work done on ENIAC but for the IAS machine which, basically, followed von Neumann's EDVAC design.[42] The basic idea is that of automating or, at least, mechanizing, that part of the "coding" process which concerns the tying together and combining of several smaller programs into one and so, amongst others, allow for the automation of access to and return from (closed) subroutines and the automation of loops. In short, the aim is to develop, what we would today call, a compiler for programs and to automate part of the programming. Indeed, as was later explained by George W. Patterson of Burroughs Company in a review of a short paper by Curry: '*automatic programming is anticipated by the author*' (Patterson 1957: p. 103). For instance, in an EDVAC-style machine, if some existing subroutine needs to be used at some point in a program, the coder needs to recalculate addresses in order to insert it into the program at the proper place. This is a time-consuming job and so having mechanizable procedures available for doing that was one major step forward.

Curry's attack on the problem was two-stepped:

1) *the analysis of programs into a set of basic programs.* These were intended as the ultimate constituents to be used as the basic building blocks for creating more complicated programs. Curry provided a set of 26 basic programs, which were a systematization of von Neumann's order code for EDVAC, and different mechanical procedures for analyzing more

41 The notion of abstract theory and so the formalization has quite a specific meaning in Curry's understanding but this is outside the scope of this paper.

42 A detailed study of Curry's work on program composition can be found in (De Mol, Carlé and Bullynck: 2015).

complicated problems into a set of basic programs. Such `analyser' was completely described for arithmetical programs but only partially for programs that also involved discrimination and loops.

2) *the (re)synthesis of basic programs into one machine-executable program.* For this Curry developed various types of substitution as well as the procedures to calculate the addresses for those substitutions. Fig. 2 gives a graphical representation of Curry's analysis of so-called simple substitution. Different operations needed in these substitution operations were described by Curry also in terms of his combinators.[43]

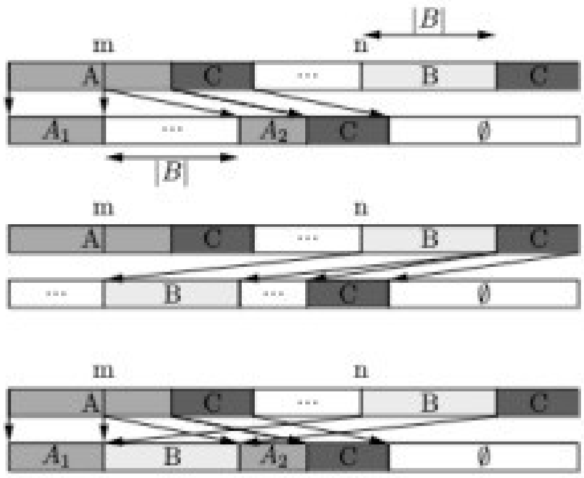

Fig. 1: Graphical representation of Curry's operation of simple substitution. This comes down to the following. Given two programs A and B which share the same data C and which are stored at different address locations, `compose' a new program ABC.

By reconnecting the earlier work of Curry on combinatory logic and his work on program composition, one sees how his earlier work on substition can effectively be reinterpreted as a dynamical model once it is (intended to be) applied effectively to program code which, by its very nature, is intended to structure a (computational) process. Viz. the potentially dynamical nature of Curry's earlier model becomes real when it is executed on a machine. Moreover, since it is the work of the human operator which is being automated, the simulative aspect of Curry's work is not about physical processes but about executing a model of human work. It is the logic of combinations and

43 Based on that analysis, Curry proposed some fundamental problems for programs such as the problem that different but equivalent algebraic expression might result in non-identical programs. Such problems are today still being studied in a computer science context where it is still unclear what it means for two programs to be identical. It is exactly in this more theoretical setting that a formal notion of simulation and bisimulation was introduced by Milner. For a recent paper on this problem, see (Angius and Primiero 2018). It is interesting to point out that within computer science, there is a very broad usage of simulation which is quite different from a notion of simulation as used within the physical and biological sciences. For instance, there is the notion of simulation as used in the context of (theoretical) machines (e.g. Turing machines) that `simulate' others and the problem to what extent one (theoretical) machine can be said to simulate another.

Curry's approach to search for the simplest possible building blocks which allowed him to capture the dynamical aspect of programs in quite a different manner than von Neumann (See Sec. 3.3.4.).

But also Curry's work on the calculation of ballistic trajectories gives another reading of `simulation' in the early computing context which is quite related to Lehmer's earlier war work on bombing patterns (Sec. 3.1.1.). It shows that the fundamental change in the use of electronic computing machinery does not so much lie in the invention of a fundamentally new scientific method (Monte Carlo simulation) as some kind of `tertium quid' (Galison 2011: 137) next to more theoretical and experimental approaches (mostly at the time) but instead in the need to rethink existing calculatory methods which were already quite commonly used in a practice which aimed at developing ways to `approximate' physical realities to test and develop artillery weapons. From that historical perspective, the idea of Monte Carlo being `*elevated […] above the lowly status of a mere numerical calculation*' (Galison 2011: 119) becomes quite problematic. It was just part of a broader historical development in science whereby one relied on calculation rather than on analytical methods or experimentation because of various reasons.

3.3. Von Neumann – formalism and mathematical physics

3.3.1. Background and involvement with ENIAC

Unlike Curry and Lehmer, von Neumann is a much celebrated computer pioneer and mathematician who has contributed (to the foundations of) a great variety of fields and subfields including computing, operator theory, economical theory, set theory and quantum theory. The following quote by Halmos gives an indication of his status amongst mathematicians (Halmos 1973: 394):

> "Von Neumann's greatness was the human kind. We can all think clearly, more or less, some of the time, but von Neumann's clarity of thought was orders of magnitude greater than that of most of us, all the time."

In other words, and to stay with (Halmos 1973), von Neumann has become a kind of hero for those involved in the disciplines to which he contributed.[44] As a consequence, one should be careful when handling his scientific biography.[45] One of the reasons for this was his apparent speed of thinking and his ability to shift so easily from one to the other domain.[46]

Despite his diversity of interests, it seems fair to say that von Neumann was first and foremost a Göttingen minded mathematician. In the early 1920s he had arrived in Berlin where he

44 As is often the case when such heroes are being created, there is also an aspect of legend (see again (Halmos 1973)).

45 So, for instance, as is pointed out in (Hashagen …), many of the existing biographies often ignore, do not engage or make contradictory claims about von Neumann's early scientific biography. In that same paper it is shown, amongst others, that around 1927 in Berlin, von Neumann was certainly not perceived of yet as a genius.

46 Because of von Neumann's status, much more has been written already on his contributions to computing and other fields and I refer the interested reader to some of the existing historical literature (Haigh, Priestley, Rope 2016; Priestley forthcoming; Aspray 1990).

met Erhard Schmidt, another Göttingen mathematician and PhD student of Hilbert, who very much supported von Neumann and discussed with him questions of foundations of mathematics.[47] It was upon the initiative of Hilbert and helped by Richard Courant that arrangements were made for him to come to Götttingen in 1926 when he was only 23 years old, to become a Rockefeller fellow (Reid 1986: 336).[48] He stayed there for the winter semester of 1926/1927 to return to Berlin. After his Habilitation in 1927 at the Friedrich-Wilhelms-Universität, he became a Privatdozent in Berlin until his emigration to the U.S.

Hilbert is of course well-known for his formalist program which, very roughly speaking, aimed at providing a consistent foundation for mathematics by relying on finitary and formalist methods.[49] Von Neumann, apparently, had a special talent for the "art" of formalization – both in providing clear and precise axioms and foundations as well as to `*think formally*´ (Ulam 1958:12) – and so it is not surprising that Hilbert very much appreciated von Neumann's work in this context.[50] However, after he had heard Gödel´s talk in 1930 at Königsberg, which presented the now (in)famous incompleteness theorem, and which is often considered as a fundamental blow to Hilbert ´s program,[51] von Neumann turned his back on the Hilbertean ideal of mathematical logic. As was the case with Curry, it would be exactly this more logical and formal work that would prove very useful for his work in computing (See Sec. 3.3.3.). It was around the same time that von Neumann was offered a position by Veblen at Princeton and so he emigrated to the United States where he would stay for the rest of his (relatively short) life.[52]

But having turned his back on mathematical logic and Germany certainly did not mean turning his back on a Göttingen tradition of establishing bridges between pure and applied mathematics and, especially, physics. In the obituary Ulam wrote for von Neumann, he points at the fact that in filling out a questionnaire of the National Science Foundation which asked, amongst others, for what von Neumann considered to be his most important contributions, he did not mention his contributions to mathematical logic but instead his contributions to the mathematical foundations of `*Quantum Theory and the Ergodic Theorem [and] the Theory of Operators*´ (Ulam 1958: 21). Ulam continues:

> This choice, or rather restriction, might appear curious to most mathematicians […] It seems to indicate that perhaps his main desire and one of his strongest motivations was to help re-establish the role of mathematics on a *conceptual level* in theoretical physics.

47 See (Hashagen 2009) for a critical discussion of von Neumann's early career and the signifance of Schmidt's support.

48 For an overview of von Neumann's early scientific biography, see (Hashagen 2006).

49 See e.g (Mancosu et al 2009) for more details.

50 See (Murawski 2004) for a discussion of von Neumann's contributions in this context.

51 It proves, very roughly speaking, that no finite axiomatic basis can completely capture mathematics,

52 Von Neumann died of cancer in 1957 when he was only 53 years old.

From the Göttingen perspective this is certainly not a curious choice. Towards the end of the 19^{th} century, Felix Klein who had actively started to transform the Göttingen mathematical department into one of the world's most famous mathematical centers, promoted a view of a close alliance between pure and applied mathematics. He arranged for cooperations with industry and also established Carl Runge as the first professor in applied mathematics at a German University. Thus, Göttingen became a centre which promoted a universalist view in which mathematics is not isolated from other sciences.[53] This Kleinean ideal was picked up or promoted at other German universities. In Berlin it was Erhard Schmidt who created a new institute for Applied Mathematics at the Wilhelm-Friedrichs University with Ludwig von Mises as a director. It is from this perspective that one can understand how Göttingen became a leading center for foundations of quantum mechanics in the 1920s with contributions from Werner Heisenberg, Max Born and also Hilbert. Runge's son-in-law, Richard Courant, was the main author of the very well-known and two-volumed Courant-Hilbert book on methods of mathematical physics which is today still a main source for mathematical physics. It was translated to English in 1938 and formed an important impetus to the formation of U.S. applied mathematics in the 1940s (Siegmund-Schultz 2009). In the introduction to that book, the view of a close alliance between the pure and applied becomes a major motivation:

> "Since the seventeenth century, physical intuition has served as a vital source for mathematical problems and methods. Recent trends and fashions have, however, weakened the connection between mathematics and physics; mathematicians, turning away from the roots of mathematics in intuition, have concentrated on refinement and emphasized the postulational side of mathematics, and at times have overlooked the unity of their science with physics and other fields. In many cases, the physicists have ceased to appreciate the attitudes of mathematicians. This rift is unquestionably a serious threat to science as a whole; the broad stream of scientific development may split into smaller and smaller rivulets and dry out. It seems therefore important to direct our efforts toward reuniting divergent trends by clarifying the common features and interconnections of many distinct and diverse scientific facts" (p. v-vi)

Courant himself was a very strong supporter of this idea. He was convinced that the pure and applied cannot and should not be separated but belong together in a unified mathematical science (Reid 1986:342). It is within that spirit that he started to investigate the mathematical significance of a method that had long been used within calculatory practices of table making, viz. the finite difference method and which resulted into the very basic and influential paper for 20^{th} century numerical analysis (Courant, Friedrichs and Lewy 1928).

53 For more details on this Göttingen tradition see (Rowe 2018).

By the time von Neumann moved to the United States, he was thus certainly familiar with the idea of bringing closer pure and applied mathematics and especially physics (cfr his work on the foundations of quantum mechanics) and so he was well equipped to become part of the small but growing community of U.S. mathematicians like Curry and Lehmer who could use their talents in the war effort in the Applied Mathematics Panel (see also (Aspray 1990: 34)). However, his work in the 1930s was still mostly within "pure" mathematics and oriented towards providing foundations for physics.

It was most probably under the initiative of Veblen (Aspray 1990, p. 26) that von Neumann was asked as a consultant in 1937 at the BRL at Aberdeen where he would become a regular visitor from that time onward. As he recounted later, it was by Robert Kent, a senior BRL official, that he was introduced to `*military science, and it was through military science that I was introduced to applied sciences. Before this, I was […] essentially a pure mathematician. […] I have certainly succeeded in losing my purity*.' (von Neumann, quoted in (Aspray 1990: 26)). At that time, von Neumann shifted attention to problems of shock wave theory and fluid dynamics and realized that the best approach to these problems might be the use of brute-force computational methods – methods which were already extensively applied at BRL at the time (See Sec. 2 and 3.2.1).

In the 1940s he also became a consultant and later a member of the Division 8 Explosives of the NDRC where he worked on problems of detonation. It is perhaps no coincidence that also Courant, who had emigrated to the U.S. after he had fled from Germany in 1933 because he was Jewish, was working on these problems. He also had been a councilor for Division 8 and then became the representative at the AMP for the contracts OEMsr-944, *Investigation in shock wave theory*, and OEMsr-945 *Research in problems of the dynamics of compressible gases, hydrodynamics, thermodynamics, acoustics, and related problems (*(Bush, Conant and Weaver 1946a). Von Neumann on the other hand was the technical representative for the contract OEMsr-1111 of the AMP with the Princeton Institute for Advanced Studies which had as a topic: `*Studies of the potentialities of general-purpose computing equipment, and research in shock wave theory, with emphasis upon the use of machine computation.*' In other words, he became the Army's specialist for the use of computing machinery and its potential for studying problems in shock wave theory. As explained, von Neumann had become convinced (or, better, had been convinced) that such problems could not be handled by the usual methods of analysis and so he proposed the use of brute-force computation instead. Indeed, in one of the summary reports of the AMP titled *Mathematical studies relating to military physical research*, a proposed numerical method due to von Neumann (von Neumann 1944) for the handling of problems of shock waves is described as follows (Bush, Conant and Weaver 1946a, p. 38):[54]

54 See (Galison 2011) for a detailed discussion of these computations.

> “The hypotheses of isentropy and that of all shocks being "straight" are generally not fulfilled. When they are abandoned, however, any exact mathematical analysis becomes quite interactable [sic], by any methods thus far employed save in exceptional cases. Consequently, considerable importance attaches to a computational treatment, developed under the Applied Mathematics Panel, which ignores shocks but which appears to produce arbitrarily good approximations to a rigorous theory allowing for shocks. The treatment depends on a much simplified quasimolecular model in place of the continuous theory. Experimental punch-card solutions produced satisfactory results with respect to precision and duration of the computations.”

Thus, in the early 1940s, von Neumann was already quite familiar not just with the idea of using brute-force to attack certain problems of (applied) physics but also with the idea of developing another model so that the numerical method could be used. Moreover, and as was the case for Curry and Lehmer, this kind of work is part of a broader and organized effort of the U.S.military and some leading mathematicians to bridge the gap between applied and pure math.

It was also because of his `*interest in explosives was genuine’* (Kistiakowsky 1980: 62) that von Neumann became a Los Alamos consultant and so he became a high-placed scientist who gained access to highly secret projects. It was in this role as an army consultant that he also paid regular visits to Aberdeen and legend goes that it was during one of those visits in 1944 that he accidentally met Herman H. Goldstine at Aberdeen Railway station where Goldstine informed hin about the highly secret ENIAC project. Goldstine arranged the clearance documents and so soon von Neumann became involved with the ENIAC project (and Goldstine his collaborator).

3.3.2. Developing mathematics on the ENIAC: the A and H bomb problems

The story about von Neumann’s work on and with ENIAC has been told a number of times and so I keep this section short.

Von Neumann’s most well-known work on ENIAC, in collaboration with several others, concerns:

(1) *the rewiring of the ENIAC*

(2) *the “first” Monte Carlo computations on ENIAC*

ENIAC rewiring The conversion of ENIAC into an EDVAC-like machine was very much driven by von Neumann and the ideas elaborated in the EDVAC report written in 1945, though he was certainly not the only one to have contributed to it. The main idea behind that conversion was to “program” the machine by instructions rather than by wiring which meant that the different instructions to be used were wired once and for all inside of the machine and so could instead be referred to by a code. There were at least two reasons for this rewiring of ENIAC into a kind of stored-program and serial machine. First of all, “coding” a problem on ENIAC required less time to set-up a problem than “wiring” it would. Secondly, in its original modus, the problems that could be set-up on the machine were limited by the number of available units (mostly accumulators and stepper counters) and so there was a serious restriction on the “size” of the problems that could be ran on it. By switching to coded instructions, there was basically no restriction on the length of a program. It was especially the latter problem that had to be resolved to implement the planned Monte Carlo computations which were prepared concurrently with the rewiring and largely motivated it.[55] In fact, it seems fair to conclude on the basis of the analyses given in (Neukom 2006; Haigh, Priestley and Rope 2015) that the Monte Carlo method and the ENIAC conversion (and its relation to (a) and (b)) were co-developed.

The “first” Monte Carlo computations on ENIAC These are the well-known computations which introduced the Monte Carlo method for the first time in the context of studying the behavior of neutron chain reactions for fission devices. Here is von Neumann’s often quoted description of the general idea behind the computation, in a letter to Richtmeyer dated March 11, 1947 and which also includes a more detailed “computation sheet” for the computation (quoted from (Metropolis 1987: 127)):

> “Consider a spherical core of fissionable material surrounded by a shell of tamper material. Assume some initial distribution of neutrons in space and in velocity but ignore radiative and hydrodynamic effects. The idea is to now follow the development of a large number of individual neutron chains as a consequence of scattering, absorption, fission and escape. [...] [A] genealogical history of an individual neutron is developed. The process is repeated for other neutrons until a statistically valid picture is generated. [...] How are the various decisions made? To start with, the computer must have a source of uniformly distributed pseudo-random numbers.”

Whereas the use of random decisions for studying (models of) physical processes in a computational or other setting, was certainly not new (see the Lehmer example in Sec. 3.1.1.) it was only with the high-speed and converted ENIAC that such methods could be more fully explored, especially for larger problems which were computation-intensive and required a relatively complex and lengthy program. The method for handling the logical complexity on the programming level was the use of a (rather complicated) flowdiagram as developed in (Goldstine and von Neumann 1947/1948).[56] Thus, we see that the computational challenges posed by the Monte Carlo computation were practically handled by the (converted) ENIAC and more systematically by von Neumann’s report on the EDVAC and the sequence of reports, written with Goldstine, on flowdiagrams (Goldstine and von Neumann 1947/1948). Thus, one can conclude that the Monte

55 Chapters 7 and 8 of (Haigh, Priestley and Rope 2015) give a detailed account of this.

56 In (Haigh, Priestley and Rope 2015) the different flowdigrans for the the different stages of the Monte Carlo computation are given and discussed in more detail.

Carlo method in its first application in the ENIAC context, not only resulted from but also very much drove the new technology under development.

The rewiring of ENIAC and the Monte Carlo computations were certainly not the only achievements to which von Neumann contributed in the ENIAC context. One interesting contribution is related to the need for random numbers for the Monte Carlo computation. Given the speed of the computations, providing the random numbers externally would slow down the process and so von Neumann came up with the idea of having the machine compute its own random numbers and so developed a numerical method for computing (pseudo)random numbers. This is the middle square method. Von Neumann's interest in pseudo-random numbers most probably led to von Neumann's interest in computing the first 2000 digits of *pi* and *e* though it should be pointed out that he was not the first one to have thought of this kind of computations in the ENIAC context (See Sec. 3.2.2.). As is recounted in (Reitwiesner 1950: 11):

> "Early in June, 1949, Professor John von Neumann expressed an interest in the possibility that the ENIAC might sometime be employed to determine the value of π and e to many decimal places with a view toward obtaining a statistical measure of the randomness of distribution of the digits [...]"

Indeed, the purpose of those computations was to know more of the probability distribution of the digits of both numbers. It was concluded that `*the material has failed to disclose any significant deviations from randomness for π, but it has indicated quite serious ones for e.*' (Metropolis, von Neumann and Reitwiesner 1950: 109) Thus, ENIAC not only made possible the Monte Carlo study on a scale that was not possible before but it also resulted in an interest in a new class of problems, viz. that of pseudo-random numbers.

Besides, I briefly mention von Neumann's contributions to numerical analysis and, more specifically, the study of error propagation, which is closely related to von Neumann's interest in pseudo-random numbers.[57] This is elaborated in the paper (Goldstine and von Neumann 1947) which provides a `*rigorous discussion*' of the problem of deriving `*rigorous estimates in connection with the inversion of matrices of higher order*' (von Neumann and Goldstine 1947: 1022) Just as Schoenberg's method of splines, it is very much rooted in the problem of round-off errors: given the discrete nature of machines like ENIAC one can `achieve any desired precision'. (von Neumann 1958: 25). However, because of the high-speed of the operations on numbers, errors occurring in each operation are superposed. Hence, it is quite important to have an estimate of the precision needed in order to avoid such round-off errors.

57 In fact it is quite probable that von Neumann's so-called middle-square method originated in his research on round-off errors. See in this context (von Neumann 1951).

3.3.3. A Göttingen mathematician engaging with ENIAC - reflections

The shift in von Neumann's work around 1940 from developing pure mathematics for theoretical physics to the use of brute-force calculations might seem curious at first. Indeed, even today *`Proceeding by "brute force" is considered by some to be more lowbrow'*. (Ulam 1980:94). As is shown in Sec. 3.3.1. however, von Neumann was already coming from a tradition where the building of bridges between the pure and applied was very much promoted. Also, von Neumann himself became an active supporter himself of this viewpoint. The following quote from *The mathematician* (von Neumann 1947b) in fact echoes some of the words from the Courant-Hilbert book quoted in Sec. 3.3.1.:

> I think that it is a relatively good approximation to truth […] that mathematical ideas originate in empirics, although the genealogy is sometimes long and obscure. But, once they are so conceived, the subject begins to live a peculiar life of its own and is better compared to a creative one, governed by almost entirely aesthetical motivations […] As a mathematical discipline travels far from its empirical source [...] it is beset with very grave dangers. It becomes more and more purely aestheticizing, more and more purely I'art pour I'art. [...] *But there is a grave danger that the subject […] so far from its source, will separate into a multitude of insignificant branches, and that the discipline will become a disorganized mass of details and complexiti*es [m.i.] […] [W]henever this stage is reached, the only remedy seems to me to be the rejuvenating return to the source: the re-injection of more or less directly empirical ideas.

So, when von Neumann was introduced at BRL and saw the potential of using calculation for problems of applied mathematics at work, the move from the pure to the applied side is perhaps less surprising. The computer then, both as a tool for studying problems from mathematical physics and as a device which demands its own mathematical theory, is exactly the kind of in-between which served very well this purpose of re-injecting empirical ideas into mathematics. And an in-between it certainly was to von Neumann. In one of his lectures of the series of lectures titled *Theory and Organization of Complicated Automata* delivered at the University of Illinois in December, 1949, he frames computing machines as tools in-between mathematical and experimental methods for studying certain problems (typically, the non-linear ones like turbulence) (von Neumann 1966, pp. 33–35):

> "In pure mathematics the really powerful methods are only effective when one already has some intuitive connection with the subject […] A very great difficulty in any new kind of mathematics is that there is a vicious circle: you are at a terrible disadvantage in applying the proper pure mathematical methods unless you already have a reasonably intuitive heuristic relation to the subject and unless you have had some substantive mathematical

> successes in it already [...] progress has an autocatalytic feature. Almost all of the correct mathematical surmises in [the area of the non-linear sciences] have come in a very hybrid manner from experimentation. If one could calculate solutions in certain critical situations […] one would probably get much better heuristic ideas. [...] there are large areas in pure mathematics where we are blocked by a peculiar inter-relation of rigor and intuitive insight, each of which is needed for the other, and where the unmathematical process of experimentation with physical problems has produced almost the only progress which has been made. *Computing, which is not too mathematical either in the traditional sense but is still closer to the central area of mathematics than this sort of experimentation is, might be a more flexible and more adequate tool in these areas than experimentation.*[m.i.]"

Thus, for von Neumann, the computer was the ideal tool to study those problems for which the more traditional methods of mathematics did not allow for an intuitive understanding of the problem set and so the computer, which was still more mathematical than pure experimentation, was the next best thing available. We see here von Neumann's version of Galison's "tertium quid" (Galison 2011: 137): the computer as a tool which connects the two traditions of experimental and mathematical physics. However, as was the case with Curry and Lehmer too, this `tertium quid' was very much prepared by a tradition in which one was already moving across disciplinary boundaries.

Von Neumann's reflections on computing machines *as a user* went hand-in-hand with reflections from the more theoretical side of the bridge between the pure and applied: there is of course his later work on automata, which provided a basic model used in current simulation contexts viz. cellular automata, but before that he also described a more theoretical model for a computing machine known as the EDVAC model and which was very much rooted in some of the issues with the original ENIAC (see Sec. 3.3.2.). As has been discussed elsewhere (Aspray 1990; Haigh, Priestley, Rope 2016), that model abstracts from engineering details and instead uses the formal model of neuron nets developed by McCullogh and Pitts (McCullogh and Pitts 1943). The latter paper also abstracts away from `*the physiological and chemical complexities of what a neuron really is*' (von Neumann 1966: 43) and instead uses a framework of formal logic. Amongst others, it refers to Turing's work on abstract computing machines, viz. formal models for defining the notion of (human) computability and which retrospectively became known as an important model for the modern computer[58] The McCullogh and Pitts paper thus had an important effect on von Neumann's work in computing not just by the application of its methods and ideas in the EDVAC context, but also as a basic reference in his reflections on natural and artificial automata.[59] This is not surprising:

58 But see (Daylight2014;Haigh 2014) for a critical discussion of the significance of the model for the development of the modern computer.

59 See for instance the second lecture at Illinois University titled `Rigorous Theories of Information and Control' which deals mostly with the Turing model and the McCullogh-Pitts model.

the idea of connecting the field he had turned his back on, viz. mathematical logic, with `empirical' processes, whether they be engineered or not, must have been very appealing to him in the light of his viewpoint as phrased in *The mathematician*: it offered him yet another opportunity to reinject the empirical back into that part of mathematics which had been so devoid of any empirical content and, perhaps because of that, failed (at least in von Neumann's view).[60]

The possibility of a new role for formal logic within the field of computing was further explored by von Neumann in his work on programming (together with Hermann H. Goldstine) and developed in the three reports (von Neumann and Goldstine 1947-48).[61] These reports are often considered as one of the first historical sources on programming:[62] the flowdiagram idea developed in it, was for a long time a much used method within programming.[63] According to von Neumann, the need for a more logical approach to programming, which he called the planning and coding of problems,[64] is rooted in the high speed of the computer (von Neumann 1948):

> "[C]ontemplate the prospect of locking twenty people for two years during which they would be steadily performing computations. And you must give them such explicit instructions at the time of incarceration that at the end of two years you could return and obtain the correct result for your lengthy problem! This dramatizes the necessity for high planning, foresight, and consideration of the logical nature of computation. *This integration of logic in the problem is a consequence of the high speed.* [m.i.]"

More specifically, logic can be used to structurally capture the so-called `dynamical' aspect, the flow, of a computation – that is, the structured ordering of how orders have to executed. It is introduced mostly because of the use of loops, subroutines and conditionals in high-speed computing which make that the actual flow of the program is not reflected by the mere sequencing of coded instructions, viz. the control C might have to jump back or forward. This dynamical stage of the planning and coding of problems was quite strictly separated by von Neumann from the so-called static part of coding or the microscopic stage of coding, viz. the individual coding of every single operation indicated in the flowdiagram. Indeed, the flowdiagram is the tool to be used for the

60 In fact, it is worth pointing out that in *The Mathematician* von Neumann's main argument in favor of reconnecting the pure with the empirical is concerned with Gödel's incompleteness results which showed him that one should be cautious `*against taking the immovable rigour of mathematics too much for granted. This happened in our own lifetime, and I know myself how humiliatingly easily my own views regarding the absolute mathematical truth changed during this episode'* (von Neumann 1947)

61 The following paragraphs are based on two talks I gave in 2014 and 2016. The slides from 2016 (in French) are available here: https://hal.archives-ouvertes.fr/cel-01345599

62 But see (De Mol, Carlé and Bullynck 2015) for a critical discussion.

63 See (Ensmenger 2016) for a historical paper on the use of flowdiagrams (later called flowcharts)..

64 It is quite interesting to point out that von Neumann did not really use the programming terminology which, as is known, had its roots in ENIAC (See (Grier 1996) for a short paper on the use of this programming technology in the ENIAC. It should be pointed out however that the word"program" was effectively used by, amongst others, Curry to refer not just to the circuitry but also to the more contemporary understanding of `program'.

dynamic stage whereas the static stage is merely a sequence of coded instructions resulting from “using” the information of the flowdiagram (See Fig. 3 and 4).

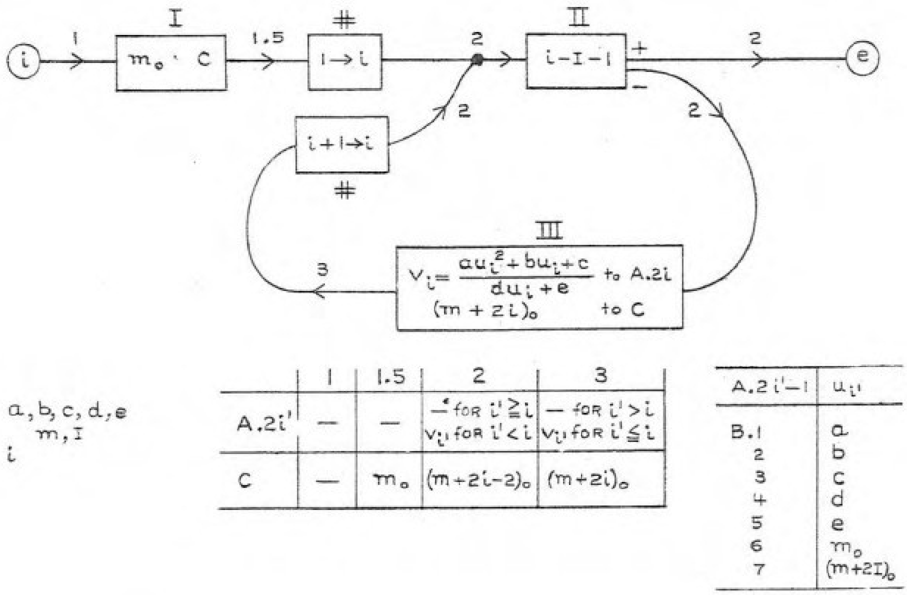

Fig. 3: Flowdiagram of the first problem discussed in (von Neumann and Goldstine 1946-48) which requires a flowdiagram. This is the following problem: for $u_1, ..., u_i$ compute $v_1, ..., v_I$ with $v_j = a^2u_j + bu_j + c/ du_j + e$ with each pair u_j, v_j stored at two neighbouring locations m + 2j-2, m+2j-1 and m and I stored at certain memory locations.

This separation between the static and the dynamical stage in the planning and coding is quite strict and reflected in what one could call a division of labor for the planning and coding of programs where the dynamical stage requires a `*mathematician or [..] [a] moderately mathematically trained person*’ and the static stage someone with a `*moderate amount of experience*’.[65]

III,1	C		Ac	$(m + 2i-2)_0$
2	III,3	Sp	III,3	$(m + 2i-2)_0$
3	-			
[	m + 2i-2	]	Ac	u_i
4	s.1	S	s.1	u_i
5	C		Ac	$(m + 2i-2)_0$
6	B.8	h	Ac	$(m + 2i-1)_0$
7	III,22	Sp	III,22	m + 2i-1 S
8	B.8	h	Ac	$(m + 2i)_0$
9	C	S	C	$(m + 2i)_0$
10	B.4	R	R	d
11	s.1	x	Ac	du_i
12	B.5	h	Ac	$du_i + e$
13	s.2	S	s.2	$du_i + e$
14	B.1	R	R	a
15	s.1	x	Ac	au_i
16	B.2	h	Ac	$au_i + b$
17	s.3	S	s.3	$au_i + b$
18	s.3	R	R	$au_i + b$
19	s.1	x	Ac	$au_i^2 + bu_i$
20	B.3	h	Ac	$au_i^2 + bu_i + c$
21	s.2	÷	R	$v_i = \frac{au_i^2 + bu_i + c}{du_i + e}$

65 There are in fa[illegible] lstin 1947-48):
Stage 1: The [illegible] ization as well as an erro[illegible]
Stage 2: the d[illegible] *lly trained person* [illegible]
Stage 3: the s[illegible]
Stage 4: the fi[illegible] *s to their final numbers.* No indication on who should do this.

Fig. 4: Part of the static coding for the problem from Fig. 3.

The result is a framework which *in theory* looks very nice but in practice becomes quite quickly a rather messy method. Indeed, even in the first problem which requires a flowdiagram (Fig. 3), one quickly sees the need for adding additional information outside of the flowdiagram (e.g. for the memory allocation). Moreover, the third volume of the report, which was assumed to deal with the problem of subroutines and which is basically the same as Curry's problem of program composition, provides an approach which is very close to the machine and certainly does not allow for the kind of automation Curry was aiming at. Nonetheless, von Neumann was convinced that (von Neumann 1948):

> "the problem of coding routines need not and should not be a dominant difficulty [In] fact we have made a careful analysis of this question and we have concluded from it that the problem of coding can be dealt with in a very satisfactory way."

And indeed, later he would not revisit this coding problem but focus instead on, for instance, the memory problem which he considered to be unresolved. The basic difference between Curry and von Neumann is perhaps the fact that in von Neumann's understanding a "program" is a representation of the dynamical behavior of the machine whereas in Curry's case it is much more about the program itself. To quote Lehmer, who was apparently not aware of Curry's work (Lehmer 1951):

> "Much has been said, but little written, about the logic or even the topology of programming. Logicians and topologists are not coming to the rescue of the desperate programmer. [...] This is the combinatorial complexity to which I have referred. Flow diagrams showing the routines, subroutines,and other wheels within wheels are hardly distinguishable from the block diagrams of the machine itself; the latter, however, are made once and for all. This then is the white mans burden of large-scale computing."

The result is a static schematic which cannot be used in a dynamical manner by the machine, or, put differently, in contrast to Curry's theory of program composition, von Neumann's model of the planning and coding of problems is not a dynamical one and so cannot result in a `simulation' of the programmer/coder and so the planning remains separated from the coding itself.

4. Discussion `A pretence of what is not'

What, if anything, is the impact of high-speed computing on science? This is the driving question behind current discussions on computer simulation. The aim of this paper was *not* to contribute *directly* to such discussion but instead to provide *a broader and deepened historical perspective* by revisiting the so-called roots of computer simulation in the use of Monte Carlo methods by von Neumann and Ulam and studying the historical context in which those arose.

As is shown here the Monte Carlo computations belong to a broader (set of) institutional, scientific and technological traditions. Thus, the apparent *`chaotic assemblage of disciplines*' which *`have no single history that can be narrated smoothly across time'* but which shared a practice that was *`sufficiently congruent in the years just after World War II*' (Galion 2011: 119) becomes less chaotic in the light of the steady organization of military science and the involvement of mathematicians. This goes back, for the U.S. at least,[66] to the end of the first World War and so when the U.S. enters the second World War, several mathematicians like Lehmer and Curry get involved with the war effort. Moreover, there is von Neumann (and with him several other emigrated mathematicians) who was coming from the Göttingen tradition which promoted a view of a close connection between pure and applied mathematics. Finally, there were also the existing calculational practices of table making for scientific or other purposes and which involved the working together of mathematicians, engineers and human computers as well as the increased use of calculatory instruments. It is against this background that one can understand that when the AMP is created in late 1942, there is a *`complete lack of astonishment which greeted [the] contributions [of the AMP members]'* (Bush, Conant, Weaver 1946a, p. v). Or, put differently, the idea of *`radically different activities*' which could now be locally coordinated in the trading zone that was supposedly constituted by Monte Carlo (Galison 2011: 119), needs perhaps to be seen in the inverse: it is the steady working together and crossing of disciplinary boundaries together with the equally steady mechanization and organization of computation which makes possible and requires the development of new methods and technologies. These new methods and technologies in their turn, increased the need for this kind of crossing of (perhaps retrospectively so-construed) disciplinary boundaries. They did not, however, make it possible.

But the historical perspective on Monte Carlo not only needs to be broadened by connecting it to what came before, and so as the result and continuation of a development, but also in relation to what happened around the same time locally. First of all, thé Monte Carlo computations were clearly not the first or only instances in which one decided to rely on calculation rather than on

66 The U.s: was actually quite late with this in comparison to, for instance, France. See (Aubin and Goldstein 2014).

"pure" mathematical methods and/or "real" experiments at that time. Perhaps the most explicit example of this discussed in this paper was the use of so-called model experiments which are very close if not identical to the idea behind the Monte Carlo computations.[67] Moreover, and especially if we consider the ENIAC context, it is understood that the main concern was the development of efficient mathematical methods and programming techniques which were adapted to the particularities of the machine (its high-speed and general programmability) in order to tackle older and newer problems. One example is the Curry inverse interpolation problem which is framed in a problem context where one aimed at `approximating' physical realities to test and develop artillery weapons. Put differently, it is the machine and the problems one wanted to attack with it which required a fundamental rethinking of the three interrelated practices around the machine, viz.:

- The mathematics used and developed, see for instance the use of an idiot approach (Sec. 3.1.2.); new approximation methods (Sec. 3.2.2.) or new methods for generating "random" numbers (Sec. 3.3.2.)
- Programming, see for instance Curry's theory of program composition (Sec. 3.2.3.) or von Neumann and Goldstine's flowdiagram approach (Sec. 3.3.3.)
- machine design, see for instance the ENIAC conversion and EDVAC design (Sec. 3.3.3.)

These structured practices are given content through the particular viewpoints, backgrounds and purposes of the different "users" involved. Thus, what mattered first of all was not questions of `simulation' but questions of good computational methods for specific problems independent of whether or not they created some kind of tertium quid. In that perspective, the Monte Carlo technique is just one of the many methods and there is no fundamental reason why it should be considered as being more `elevated' or special than say Lehmer's exponent computation or Schoenberg's splines.

Moreover, we see that if one re-interprets retrospectively some of the other work that was done in the wake of the ENIAC context, as simulative work, that the usual understanding of `simulation in the 1940s and 50s' as in mimicking some physical process by relying on randomized methods, at least needs to be recast in the context of programming (and so consider the intertwining of mathematical logic and engineering) and the context of number theory (and so the computer as an "analog" machine for doing number theory).

Thus, instead of focusing on (philosophical issues of) simulation and on what the relation is between the simulator and what it mimicks, the focus was here on what it actually *is* and how it is *made*. By so doing, I have shown that accounts of (computer) simulation which assume a discontinuous history which has one main starting point (the rise of the modern computer and the

67 Though, of course they were not used to study thermonuclear problems nor were the computations done with the help of a high-speed device like ENIAC.

“first” Monte Carlo computations) give a reductive and singular account which might distort our current views on simulation. As such, one strengthens what can be understood as a double hiding in research on simulation. First, by focusing on simulation and not on computational methods one hides the plurality and diversity of practices and how they are determined not just by what they are supposed to simulate but also the techniques, programming interfaces and, ultimately (historically) the machine. Secondly, because of the additional connotation of `simulation’ as a “mimicking” this hiding is strengthened. Viz, it suggests that it suffices to look at what is being simulated and not at what constitutes the simulator. As such, simulation really becomes “a pretence of what is not”, connecting it back to its older meanings.[68] This is not only a problem for philosophers or historians of science but a very real one today when science more and more relies on computation and where scientists alongside mathematicians are more and more confronted with issues of transparancy, error, open and closed software, etc (See e.g. (Ince 2012)). Uncovering some of the historical complexities of `simulation’, retracing its so-called origins, is one strategy against such issues.

Bibliography

Alt, F., Interview with Franz Alt by Henry Tropp, September 12, 1972, Computer Oral History Collection, 1969-1973, 1977.

Alt, F., Archaeology of Computers. Reminiscences, 1945-1947, *Communications of the ACM*, vol. 15, nr. 7, 1972, pp. 693-694.

Alt, F., “Dr. Franz Alt. Oral history by Atsushi Akera, January 23 and February 2, 2006”, in: *Proceeding ACM Oral History interviews*, interview nr. 1, New York: ACM.

Anderl, S., Implicit changes of representational ideals in astrophysics - The example of the Paris-Durham shock model”, forthcoming.

Angius, N. and Primiero, G., 2018, The logic of identity and copy for computational artefacts, *Journal of Logic and Computation*, https://doi.org/10.1093/logcom/exy012

Archibald, T., Dumbaugh, D. and Kent, D., U.S. mathematicians in World War I, in: (Aubin and Goldstein 2014), pp. 229-271.

Aspray, B, *John von Neumann and the origins of modern computing*, MIT Press, 1990.

Aubin and C. Goldstein (eds.), *The war of guns and mathematics. Mathematical practices and communities in France and its Western Allies around World War I*, American Mathematical Society, 2014.

Borrelli, A., “Program FAKE: Monte Carlo simulations as tools of theory in early high energy physics”, forthcoming.

68 See the OED and its reference to meanings of simulation prior to 1947.

Bush, V., Conant, J.B. and Weaver, W., Mathematical studies relating to military physical research, National Defense Research Committee, Summary Technical report of the Applied Mathematics Panel, NDRC, vol. 1, AD221604, 1946a.

Bush, V., Conant, J.B. and Weaver, W., Probability and Statistical studies in warfare analysis, National Defense Research Committee, Summary Technical report of the Applied Mathematics Panel, NDRC, vol. 3, 1946b.

Barberousse, A. and Imbert, C., “Le tournant computationnel et l’innovation théorique”, in: Soazig Le Bihan (ed.), *Précis de philosophie de la physique*, Paris: Vuibert, 2013.

Bergin, T.J. (ed.), 50 years of army computing. From ENIAC to MSRC. United States Army Research Laboratory (ARL), 2000

Bredella, N., “How the widespread use of Building Modeling came about through a mesh of changes within and without technology”, forthcoming.

Bullynck, M. and De Mol, L, “Setting-up early computer programs: D. H. Lehmer’s ENIAC computation”, *Archive for Mathematical Logic*, vol. 49, 2010, pp. 123-146.

Bullynck, M., Computing Primes (1929-1949): Transformations in the Early Days of Digital Computing, *IEEE Annals of the history of computing*, vol. 37, nr. 3, 2015, pp. 44-54.

Bullynck, M. and De Mol, L., “The history (of the history) of computing and its relation to the history (of the history) of mathematics.” in: M. Bullynck, L. De Mol and M.-J- Durand-Richard (eds.) *Mathematics and Machines*, special issue of *Revue de Synthèse*, forthcoming.

Bush, V., Conant, J.B. and Connor, R.A., Summary technical report of Division 8, NDRC, vol. 1 The preparation and testing of explosives, 1946.

Campbell-Kelly, M., Croarken, M., Flood, R. and Robson, E., The history of mathematical tables. From Sumer to spreadsheets, Oxford University Press, 2003.

Campbell-Kelly, “The computation factory: de Prony’s project for making tables in the 1970s”, in: Campbell-Kelly et al, 2003, pp. 105–121.

Courant, R., Friedrichs, K. and Lewy, H., "Über die partiellen Differenzengleichungen der mathematischen Physik", Mathematische Annalen, vol. 100, nr. 1, 1928, pp. 32–74

Curry, H.B., An analysis of logical substitution, American Journal of Mathematics, vol. 51, nr. 3, pp. 363-384.

Curry, H.B., Mathematical Teaching and National Defense, The School Review, vol. 50, nr. 5, 1942, pp. 337-346.

Curry, H.B., The heaviside operational calculus, *The American Mathematical Monthly*, vol. 50, nr. 6, 1943, pp. 365-379.

Curry, H.B., The method of steepest descent for non-linear problems, Quarterly of applied mathematics, vol. 2, nr. 3, 1944, pp. 258-261.

Curry, H.B. and Wyatt, W.A., *A study of inverse interpolation of the Eniac*. Technical Report

615, Ballistic Research Laboratories, Aberdeen Proving Ground, 1946.

Curry, H.B., Schoenberg, I.J., On spline distributions and their limits: The Polya distribution functions. Bulletin of the American Mathematical Society, vol. 53, 1947, p. 1114.

Curry, H.B., Schoenberg, I.J., On P ́olya frequency functions IV: The fundamental spline functions and their limits. Journal d'Analyse Mathematique, vol. 17, 1966, pp. 71-107.

Curry, H.B., *On the composition of programs for automatic computing.* Technical Report 9805, Naval Ordnance Laboratory, 1949.

Curry, H.B., *A program composition technique as applied to inverse interpolation.* Technical Report 10337, Naval Ordnance Laboratory, 1950.

Daylight, E., *A Turing Tale*, Communications of the ACM, vol. 57, nr. 10, pp. 36-38.

De Mol, L., Doing Mathematics on the ENIAC. Von Neumann's and Lehmer's different visions, Witzke, I. (ed.), *Mathematical Practice & Development throughout History. Proceedings of the 18th Novembertagung on the History, Philosophy and Didactics of Mathematics*, 2009, pp.149--186.

De Mol, L., Some reflections on mathematics and ts relation to computer science, in: Andy Adamatzky, *Automata, Universality, Computation. Tribute to Maurice Margenstern*, Springer, 2015, pp. 75-101.

De Mol, L., Carlé, M. and Bullynck, M., Haskell before Haskell: an alternative lesson in practical logics of the ENIAC, *Journal of Logic and Computation*, vol. 25, nr. 4, 2015, pp. 1011–1046.

De Mol, L. and M.-J- Durand-Richard, "Calculating machines and numerical tables – a reciprocal history", in: Tournès, D. (ed.), *The history of numerical tables*, Springer, forthcoming, preprint available at: http://hal.univ-lille3.fr/hal-01396846

Eckert, P. J., An interview with J. Presper Eckert, conducted by Nancy Stern on 28 october 1977, Charles Babbage Institute, Retrieved from the University of Minnesota Digital Conservancy, http://hdl.handle.net/11299/107275.

Ensmenger, N., The multiple meanings of a flowchart, Information and Culture: A journal of history, vol. 51, nr. 3, pp. 321-351.

Eudey, M.W., Fix, E., Lehmann, E., Lehmer, D.H., Lehmer, E., Neyman, J., Robinson, J., Shane, M. and Scott, E., *Cooperative study on area bombing*, OEMsr-817, (Report submitted to AMP, NDRC), Statistical Laboratory, University of California, Oct. 1944, AMP-803.2-M.

Forman, P., "The Primacy of Science in Modernity, of Technology in Postmodernity, and of Ideology in the History of Technology," *History and Technology*, vol. 23, nr. 1-2, 2007, pp. 1–152.

Fritz, B., "ENIAC – a problem solver", *IEEE Annals of the history of computing*, vol. 16, nr. 1, 1994, pp. 25 – 45.

Fritz, B., The women of ENIAC, *IEEE Annals of the history of computing*, vol. 18, nr. 3, 1996, pp. 13-28.

Galison, Peter, I*mage and Logic. A material culture of microphysics*, University of Chicago Press, Chicago, 1997.

Galison, Peter. “Computer simulations and the Trading Zone” In: Gabriele Gramelsberger (ed.), *From Science to Computational Science*, Zürich: Diaphanes, 2011, pp. 118-157.

Goldstine, H.H. and von Neumann, J., Planning and coding of problems for an electronic computing instrument, Vol. 2 of Report on the Mathematical and Logical Aspects of an Electronic Computing Instrument, part I,II and III. Report prepared for U. S. Army Ord. Dept. under Contract W-36-034-ORD-7481, 1947–48.

Grier, D.A., The ENIAC, the verb “to program” and the emergence of digital computers, IEEE Annals of the History of Computing, vol. 18, nr. 1, 1996, pp. 51-55.

Grier, D.A., *When computers were human*, Princeton University Press, 2005.

Grubss, F. E., Sample criteria for testing outlying observations, *The Annals of Mathematical Statistics*, vol. 21, nr. 1, 1950, 27-58.

Haigh, T., actually, Turing did not invent the computer, Communications of the ACM, vol. 57, nr. 1, pp. 36-41.

Haigh, T, Priestley, M. and Rope, C, *ENIAC in action. Making and Remaking the modern computer*, MIT Press, 2016.

Halmos, P.R., The legend of John von Neumann, *The American mathematical monthly*, vol. 80, nr. 4, 1973, pp. 382-394.

Hartmann, S., *The world as a process: Simulations in the natural and social sciences*. In: R. Hegselmann, U. Mueller and K. G. Troitzsch (eds.), *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, Springer, pp. 77–100.

Hartree, D.R., The ENIAC, An Electronic Computing Machine, *Nature*, vol. 158, 1946, pp. 500-506.

Hartree, D.R., *Calculating instruments and machines*. University of Illinois Press, 1949.

Hashagen, U., 2006a. Johann Ludwig Neumann von Margitta (1903–1957). Teil 1: Lehrjahre eines jüdischen Mathematikers während der Zeit der Weimarer Republik. Informatik-Spektrum 29, 133–141

Hashagen, U., 2006b. Johann Ludwig Neumann von Margitta (1903–1957). Teil 2: Ein Privatdozent auf dem Weg von Berlin nach Princeton. Informatik-Spektrum 29, 227–236.

Hashage, U., 2009, `Die Habilitation von John von Neumann an der Friedrich-Wilhelms-Universität in Berlin: Urteile über einen ungarisch-jüdischen Mathematiker in Deutschland im Jahr 1927,’ Historia Mathematica, vol. 37, pp. 242-280.

Hashagen, U., “The computation of nature, or: does the computer drive science and technology?”, in: Bonizzoni P., Brattka V., Löwe B. (eds),*The Nature of Computation. Logic, Algorithms, Applications. CiE 2013*. Lecture Notes in Computer Science, vol 7921. Springer, Berlin, Heidelberg, 2013, pp. 263–270.

Heim, S.J., *John von Neumann and Norbert Wiener: From Mathematics to the Technologies of Life and Death*, MIT Press, 1982.

Hoff, Frederick C., Ordnance – the backbone of warfare, Alumni Review. California Institute of Technology, vol. 6, nr. 4, 1943, pp.

Humphreys, Paul W. 2004. *Extending ourselves: Computational science, empiricism, and scientific method.* Oxford University Press.

Ince, D.C., Hatton, L. and Graham-Cumming, J., The case for open computer programs, *Nature*, vol. 482, pp. 485 – 488.

Kittler, F.A., Es gibt keine Software, in: Friedrich Kittler, *Draculas Vermächtnis. Technische Schriften*, Reclam, 1993, pp. 225-242.

Kline, R.R., Inventing an Analog Past and a Digital Future in Computing, Haigh, T. (ed.), *Exploring the Early Digital*, Springer Verlag, forthcoming.

Kistiakowsky, G.B., Reminscences of wartime Los Alamos, in: Badash, L., Hirschfelder, J.O. and Broida, H.P., Reminiscences of Los Alamos 1943-1945, Reidel Publishing Company, 1980.

Lehmer, D.H., *A Photo-Electric Instrument for Rapid Bomb-Fall Analysis and Damage Prediction*, D. H. Lehmer, OSRD 5641, OEMsr-817, AMP Report 190.1R, University of California, May 1945. AMP-804.4-M

Lehmer, D.H., Mathematical methods in large scale computing units, *Proceedings of Second Symposium on Large-Scale Digital Calculating Machinery, 1949*, Cambridge, Massachussetts: Harvard University Press, 1951, pp. 141–146.

Lehmer, D.H., High-speed computing devices and their applications, in: Edwin F. Beckenbach, *Modern mathematics for the engineer,* New York: McGraw-Hill, 1956, pp. 480-497.

Lehmer, D. H. (1963). Some high-speed logic. In Experimental Arithmetic, *High Speed Computing and Mathematics*, volume 15 of Proceedings of Symposia in Applied Mathematics, pp.41–376.

Lehmer, D.H., "Computer technology applied to the theory of numbers", *Studies in Number Theory* (W.J. Leveque, ed.), vol. 6, 1969, pp. 117–151.

Lehmer, D.H., "The influence of computing on mathematical research and education" in: J. Lasalle, ed., *The influence of computing on mathematical research and education*, Proceedings of Symposia in Applied Mathematics, American Mathematical Society, Providence, 1974, 20: 3–12.

Lehmer, D.H., "A history of the sieve process", Howlett, J., Metropolis, N. and Rota, G.-C. (eds.), *A history of computing in the twentieth century*, Academia Press, New York, 1980, Proceeding of the International Research Conference on the History of Computing, Los Alamos, 1976, 445–456.

Mancosu, P., Zach, R. and Badesa, C., The Development of Mathematical Logic from Russell to Tarski, 1900 – 1935, in: Haaparanta, L. (ed.), *The Development of Modern Logic*, Oxford University Press, New York, 2009,pp. 318 – 470.

McCullogh, W. and Pitts, W., A logical calculus for the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, vol. 5, 1943, pp. 115-133.

Metropolis, N, The beginning of the Monte Carlo method, Los Alamos Science, Special Issue, Stanislaw Ulam 1909-1984, vol. 15, 1987, 125–130.

Murawski, R., John von Neumann and Hilbert's school of foundations of mathematics, *Studies in Logic, Grammar and Rhetoric*, vol. 7, nr. 20, 2004, pp. 37-55.

Neukom, H., The second life of ENIAC, *IEEE Annals of the history of computing*, vol. 28, nr. 2, 2006, pp. 4-16.

Patterson, G.W., Review of "The logic of program composition by H.B. Curry". *Journal for Symbolic Logi*c, vol. 22, 1957, 102–103.

Polachek, H., Before the ENIAC, *IEEE Annals for the History of Computing*, vol. 19, nr. 2, 1997, pp. 25-30.

Reid, C., *Hilbert – Courant*, Springer, 1986.

Rosenbloom, P.S., *On computing: the fourth great scientific disicpline*, MIT Press, 2015.

Rowe, D., *A Richer Picture of Mathematics. The Göttingen Tradition and Beyond,* Springer, 2018.

Schoenberg, I.J., *Contribution to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of anbalytic approximation formulae. COMPLETE*

Schoenberg, I. J., A brief account of my life and work, in: C. De Boor (ed.), I.J. Schoenberg. Selected Paper, Basel: Birkhaüser, 1988, vol. 1, pp. 1-10.

Seldin, J.P., CURRY, Haskell Brooks (1900-1082). *Dictionary of Modern American Philosophers*, Thoemmes Press, Bristol, England, 2005.

Seldin, J.P., Curry's work on computers in the early days of computing, *Canadian Society for History and Philosophy of Mathematics. Proceedings of the 38th annual meeting*, May 27-29, 2012, vol. 25, pp. 149-153.

Siegmund-Schultz, R., *Mathematicians fleeing from nazi Germany. Individual fates and global impact*, Princeton University Press, 2009.

Stibitz, G.R., A talk on relay computers, AMP Memo 171.1M, National Defense Research Committee, 1945.

Swade, D. "The `unerring certainty of mechanical agency': machines and table making in the nineteenth century", in: Campbell-Kelly et al, 2003, pp. 145–174.

Tedre, M., *The science of computing: shaping a discipline*, CRC Press, 2015.

Ulam, S., John von Neumann, 1903-1957, *Bulletin of the American Mathematical Society*, vol. 64, nr. 3, 1958, pp. 1-49.

Ulam, S., von Neumann: The interaction of mathematics and computing, 1980, in: Howlett, J., Metropolis, N. and Rota, G.-C. (eds.), *A history of computing in the twentieth century*, Academia Press, New York, 1980, Proceeding of the International Research Conference on the History of Computing, Los Alamos, 1976., pp. 93–99.

Varenne, F., *Du modèle à la simulation informatique*, Vrin, Paris, 2007.

Varenne, F., “Models and Simulation in the Science of Complexity,” in: M.A. Aziz-Alaoui & C. Bertelle. *From System Complexity to Emergent Properties*, Springer, 2009, pp.3-21.

von Neumann, J., `Mathematische Begründung der Quantenmechanik. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, pp. 1--57.

von Neumann, J., Proposal and Analysis of a New Numerical Method for the Treatment of Hydrodynamical Shock Problems, AMP Report 108.1R, Institute for Advanced Study, Mar. 20, 1944.

von Neumann, J., First draft of a report on the EDVAC, contract no. w-670-ORD-492, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, 1945. Also published in *IEEE Annals of the History of Computing*, Vol. 15, No. 4, 27-75, 1993.

von Neumann, J. and Goldstine H.H., “Numerical inverting of matrices of high order”, Bulletin of the American Mathematical Society, vol. 53, nr. 11, 1947, pp. 1021-1099.

von Neumann, J., “The Mathematician”, Heywood, R.B. (ed.), *The works of the Mind*, University of Chicago Press, 1947b, pp. 180–196.

von Neumann, J., “Various techniques used in connection with random digits,” Applied Math Series, Summary written by G. E. Forsythe, in: National Bureau of Standards, Vol. 12, pp. 36-38, 1951.

von Neumann, J., *The computer and the brain*, New Haven: Yale University Press, 1958.

von Neumann, J., *The general and logical theory of automata*, edited by Arthur Burks, University of Illinois Press, Urbana, London, 1966.