Eniacisms

Computations on the ENIAC: Some set-ups with various notes

Maarten Bullynck

> **Synchronization**. Every 1/5000th of a second the cycling unit emits a fundamental pattern of signals, nine trains of special pulses and a gate. The terms *pulse* and *gate* both refer to a change in voltage, either positive or negative, from some reference level. The distinction between these is their duration. [...] One of the pulses emitted by the cycling unit, the central programming pulse (CPP) is of particular importance in this discussion. This pulse, emitted once every 1/5000th of a second marks the beginning and end of a cycle. (Goldstine and Goldstine, 1946, 98)

As Nick Metropolis would later say, plugging and programming the ENIAC in its early days "was a marvelous opportunity for an indoctrination [to computing]; it was absolutely priceless, almost" (Interview, 1987). Indeed, in setting up complex computations for the ENIAC, quite some techniques and methods, as well as problems, were developed, which for a part are still current, for a part have determined research in mathematics of computation. In what follows, a more or less chronological, though not exhaustive, list of problems that were computed on the ENIAC will be given, with some notes, and perhaps with a sligtly number-theoretically biased perspective.

Most of these problems were of military interest, which implies that though the general mathematical structure of the problem is given, the actual physical constants and/or empirically defined functions to be substituted in the equations are not. Hence, these problems can be described as accurately as a particular set of differential equations, more specifically a non-linear set of differential equations, but the constants and empirical functions in these sets will have to remain unspecified.

The rare computations not of military interest that ran on the ENIAC – to my knowledge only two – are, however, of a more theoretical, mathematical nature, more specifically even, of number-theoretic nature. In these computations not only the mathematical structure of the computation is clearly delineated, but also the physical constants are known, i.e., number theory feeds parts and properties of its numerical-notational structure in its equations, thus recurring on its own tail. Their set-up on the ENIAC will then be described in more detail than the military, non-linear problems.[1]

The chronology thus runs from Thanksgiving 1946 to Thanksgiving 1950, from the prime number sieve to the squaring of the circle. These are the spare days offered as a libation to mathematics, serving the military on all other days in between. Ironically, the subtext of both these computations – an investigation into factoring and randomness – is nowadays of more economic-military interest than the mathematically rather uninspiring differential equations that ran on the ENIAC (excepting the techniques invented to check the feasibility of the H-Bomb on the ENIAC).

[1] And it must be said: I prefer number theory to meticulously transcribing partial differential equations in finite differences!

A last remark, before starting off. The chronology (1946–1950) is interrupted by a major change in the operation routine of the ENIAC. 1947–1948 the ENIAC was converted from a parallel, pluggable computer into a sequential, programmable computer, thus altering drastically the input routines. The coding of the problems before and after 1948 is consequently very different, making a small digression on the re-formatting necessary.

1 Plugging (1946–1948)

1.1 The Multiplication

> One accumulator was loaded with the number 5, then the other with the number 1,000 and the two were multiplied together. The machine showed 5,000. (McCartney, 1999, 88)

The first near trivial computation on the ENIAC – not impressing the female computers working next door – reiterates on its own central programming pulse: 5,000. One decade ring counter containing 0 000 001 000, the other 0 000 000 005, then the multiplication table schemes are invoked, transmitting back the result 0 000 005 000, without even having to use the carry-over circuit on the receiving ring counter.
One historical remark may be made: The multiplication table circuitry implements one of the oldest mathematical techniques, going as far back as ancient Egypt, or even older, ancient China, viz., multiplication tables that the pupil in arithmetics has to store in her memory. One characteristic of this basic arithmetical knowledge lacks in the circuitry, however, the thing called 'Practicam', 'Vortheilrechnung' or shortcuts, not missing in any *Rechenbuch*. No human computer would ever perform the operations on every digit of these numbers, but would simply use the shortcut, write 3 zeros after the digit 5.
Thus, whereas the ENIAC executes 10 times 10 lookups in the units digit multiplication table, filling one accumulator with 0 000 005 000, and 10 times 10 lookups in tens digit multiplication table, resulting in a second null accumulator, adding up both accumulators to the desired result, a human calculator would only have to perform one erase operation, and one 'write 5' operation on the correct digit position.

1.2 Tables of Squares, Tables of Cubes

In their introductory paper on the ENIAC, Herman and Adele Goldstine give the set-up for a simple calculation, tabulating n, n^2 and n^3 (Goldstine and Goldstine, 1946, 107–9). An easy experiment, only using "4 percent of the machine's programming capability", but also a tabulation with a long history. Tables of squares and cubes are useful in computing solutions to Fermatian problems and/or Diophantine quadratic and cubic equations, e.g., John Pell edited a table of squares up to 10,000 exactly for this reason (Pell, 1672).
The Goldstines' procedure is neatly designed, with an eye on the economical use of the machinery, having an accumulator for n (A1), n^2 (A2) and n^3 (A3) respectively. A2 is transmitted 3 times to A3, A1 2 times to A2, 3 times to A3 and the constant 1 is transmitted once to all three A's, resulting in the desired $n+1$ (A1), $(n+1)^2$ (A2) and $(n+1)^3$ (A3). All in all, this requires 8 accumulator transmissions and 3 constant signal transmissions. However, the procedure can be improved: Transmitting once from A1 to A2, thrice from A2 to A3, 2 constant transmissions to A1 and A3 and one transmission from A1 to A2, adding up to 5 accumulator transmissions and two signal transmissions. In cycles: the Goldstines' solution takes 18 CPP, our optimalized one only 13 CPP.

1.3 The Sieve

> While it was being built there in Philadelpia, under the direction of J. Mauchly, J.P. Eckert and others, the Proving Ground assembled a "Computations Committee" to prepare for utilizing the machine after its completion. The committee consisted of: Derrick H. Lehmer, the number theorist from Berkeley; Haskell B. Curry, the logician from Penn State; Leland B. Cunningham, a young astronomer who later became professor at Berkeley; and the author [Franz Alt] (Alt, 1972, 693–4)

This 'Computations Committee' represents quite accurately the various strands in the mathematical heritage, who traditionally busied themselves with extensive computations. Number theory, though having acquired an 'abstract' touch in the 19th century, traditionally depends on large tables displaying various properties of numbers, on the basis of which theorems and conjectures are formulated, and also has perhaps more near intractable conjectures than any other branch of mathematics. Since the 19th century, logic had been looking for the smallest set of simple axioms or instructions, on which one could found the rest of logic (and mathematics), thus giving rise to explosive combinatorial questions, since less elements imply more intricate and large combinations to arrive at something meaningful. Astronomy then has a long standing tradition in having to deal with 1) large observational data sets, 2) complex formulae for i) combining and reducing these data, and for ii) approximating non-integrable functions. Lastly, the tradition of building machinery to implement arithmetical operations is represented in the person of Franz Alt, who was in charge of developing G. Stibitz's relay multiplicators at Bell Labs.
If Curry, Cunningham and Alt contributed to devising large test programs, I have thus far been unable to verify, but Lehmer's test program is luckily enough well-documented be it in a fragmentary way, scattered over some essays from Lehmer's hand, written up 20 years later.

> D.H. Lehmer [...] had programmed [one of the first problems ever run to completion on an electronic computer] on ENIAC, with J. Mauchly serving as "computer operator", during the three-day weekend of July 4, 1946. (Alt, 1972, 694)

More specifically, after Lehmer's own account (Lehmer, 1976), he and his wife Emma, also a number theorist, together with their children arrived at the ENIAC, discussed on how to set up the computation, quite by accident happened on J. Mauchly, and finally went to a nearby hamburger restaurant, where the Lehmers and Mauchly devised the plugging on a paper napkin.
The problem implemented is a favourite topic of D.H. Lehmer: Sieve Problems. The most basic and oldest sieve is Eratosthenes's Sieve, where the successively found prime numbers serve as multiplicators, eliminating all composite numbers further in the series (or in a square arrangement, which is more practical) of integers. Lehmer's father, D.N. Lehmer also a number theorist, had rediscovered Gauss's method of speeding up this procedure by using the quadratic residues, hence the name quadratic sieve. Implementing this quadratic sieve and other sieves would become one of many hobbies of his son D.H. Lehmer. He devised in the 1930ies a photo-electric device to sieve numbers (Lehmer, 1933b), and described the general sieve problem for digital equipment in (Lehmer, 1953). Sieves are real number crunching problems, computationally very demanding, except when they can be implemented in parallel. Sieve problems just beg for parallel computation.
On the ENIAC ran a somewhat different though related sieve.

> The problem proposed to the ENIAC was to find those primes p for which $e(p) \leqslant 2000$ until Monday morning at 8 o'clock. During the weekend the limit 2000 was reduced to 1000 and then to 300 in order to speed things up. By Monday, we had reached $p = 4538791$. This succesful run had a number of consequences, even legal ones, which I shall not discuss. (Lehmer, 1974, 4)

The legal problem was probably the fact that the ENIAC was at that time still a classified project, and that it was forbidden to publish the obtained results.[2] The function $e(p)$ is equal to the exponent e, for which $2^e \equiv 1 \pmod{p}$, or in a different phrasing, given the number p in binary notation, how long is the period of the binary expansion of $\frac{1}{p}$? If e is equal to $p-1$, than e is a primitive root of p, which implies that p is a prime number. If e is smaller than $p-1$ (it will actually be a factor of $p-1$) than it is not a primitive root, but if e is much smaller than $p-1$, than a polynomial time algorithm can be constructed to test the primality of p, giving immediately its factors.[3]
This is thus the interest in finding p's with a small value of e. Since the ENIAC still was a parallel computer in 1946, this problem could be implemented as a sieve problem, producing 2^e and checking on the condition $\equiv 1 \pmod{p}$ for several p's at a time, sifting out the p with small e's. More details on the implementation on the ENIAC are given in (Lehmer, 1974).
[A complete reconstruction should be substituted for this parenthetic comment]
Though the sieve problem is unrelated to any military project, there as another classic of sieve problems that has some connection to the army. The main theorem used in this sieve, known as the Chinese Remainder Theorem, has been conserved in Chinese tracts in the form of a problem. The problem is to count the number of soldiers in a regiment of, say, some thousand men. Successively counting every single individual is of course very liable to error, instead one could order the regiment to arrange in groups of 25 and in groups of 49 successively. Counting the 'rest soldiers' not making up a complete group of 25 and 49, gives the clue to find the total number of soldiers. Combining the equations $x \equiv r_1 \pmod{25}$ and $x \equiv r_2 \pmod{49}$ into $x \equiv R \pmod{25.49 = 1225}$ (since 25 and 49 are relatively prime) delivers the answer, the total number is $R + 1225$.[4] Therefore an army is a parallel computing device ;-)

1.4 Firing Tables and other Non-Linear Problems

In a book review in the fascinating journal *Mathematical Tables and other Aids to Computation* (MTAC, 1943–1955), D.H. Lehmer (*MTAC* (1946) 2 (19), 317) mentions the following set of equations:

> Part 6 [of D.R. Hartree's *Calculating Machines* (1946)] is a short description of an interesting boundary layer problem which the author put on the ENIAC in June 1946. It consists in solving the non-linear system of three differential equations
> $f' = h(1+\alpha r)^{1/9}, \quad h'' = -fh', \quad \beta r'' = fr' + (h')^2$
> with the two-point boundary conditions,
> $f = h = r' = 0 \quad \text{at} \quad x = 0, \quad h = 2, \quad r = 0 \quad \text{at} \quad x = \infty$

[2] Indeed, Lehmer never published the results from this test, probably because by the time he was free to do so, they had become completely obsolete.
[3] For a more elaborate explanation and history of this 'thing', I refer to (Bullynck, 2005, October 2006 Version).
[4] See (Lehmer, 1933a).

Considering the date of the computation, this might very well be one of the test problems for the ENIAC, to test for computations of non-linear sets of (partial) differential equations.[5] Nearly all interesting and realistic problems in physics (i.e. not in an artificial set-up such as an experiment) lead to sets of non-linear differential equations. Such is the case for ballistics, the science of firing bullets or bombs. To answer the question where a bullet will end up, such sets have to be solved. Hence, Hartree's test problem may very well be say preparing the endless computations of firing tables.

It was the urgent need for firing tables during the Second World War that got the ENIAC-project funded by the U.S. army, and firing tables the ENIAC indeed computed *en masse*. The theory and computation of firing tables has a history dating back to the introduction of gunpowder in Europe, though consistent theories were only formulated in the 18th century.[6] 1940, preparing the U.S. war effort, L.S. Dederick, mathematician at the Aberdeen Proving Ground, gave a survey of the state of the art in ballistic computation.

> The mathematical theory upon which ballistic computations are based is subject to change depending upon two very different conditions. One is the kind of weapon and the consequent type of trajectory which is common at the time in actual gunnery, and the other is the available method of a strictly computational character. (Dederick, 1940, 628)

In general, the problem is given by the following differential equations:

$$\begin{aligned} x'' &= -Ex' \\ y'' &= -Ey' - g \end{aligned}$$

with g is acceleration of gravity, and $E = \frac{GH}{C}$, where G is the empirically determined drag function (depending on the shape of the projectile and with arguments x, y, v_x, v_y), H ratio of air density and C the ballistic coefficient. Working this out, assuming a trajectory in two dimensions, and putting $u = v_x sec\theta_0$ and $m = tan\theta = v_y/v_x$ the differential system becomes:

$$\begin{aligned} \frac{dt}{du} &= \frac{-1}{Eu} \\ \frac{dm}{du} &= \frac{gsec\theta_0}{Eu^2} \\ \frac{dx}{du} &= \frac{-cos\theta_0}{E} \\ \frac{dy}{du} &= \frac{-mcos\theta_0}{E} \end{aligned}$$

This non-linear system cannot be integrated with standard functions (i.e., logarithms, sines, hyperbolic sines &c.), and thus has to be approximatively computed.

Before 1918, the Siacci method was used to calculate solutions, assuming the θ very small and H nearly 1. In this way, the system could be partially integrated, and a table could be constructed giving the solutions for a specific drag function G. After 1918, hard numerical integration by the method of finite differences, and aided by manually operated

[5] Differential equations are like classic equations, only no numerical value should be found, but a function fulfilling the conditions of the equations. A set of such equations is non-linear, if they cannot be brought in a linear form $h' = f(x)$ where h' is shorthand for $dh(x)/dx$, that is, if equations of higher order (e.g. $d^2h(x)/dx^2$) are involved, or if the equations are interrelated in such a way they cannot be disentangled into simpler equations.

[6] More specifically by J.H. Lambert (1765/1767), cfr. (Szabó, 1979, 220-24).

desk calculators, was used. This method had better approximations, and was calculated for standard conditions (G and H remain unchanged). If these standard conditions changed, the numerical integration had to be done all over again, therefore a variational method, the adjoint system, was invented, so as to rapidly adjust the already obtained solutions to new conditions. With the advent in the 1930ies of mechanical integration, not manually operated but automatic calculators, this method just as the Siacci method before, became obsolete. However, as Dederick shows, with some modifications, both methods can still be used to improve the work with mechanical integrators.
The exact sets of equations that were plugged on the ENIAC cannot be reconstructed, for this implies the knowledge of the drag function G and the other constants. However, it is certain that different variations on the above set of differential equations were fed into the ENIAC, using the method of finite differences to split this non-linear system into discrete parts (in contrast to the differential analyzers, who computed continuously). On the Symposium for Large Scale Digital Computing (1948) Richard Courant presented a general overview on how to break up such a non-linear system into finite differences (Courant, 1948/1950).

1.5 Classified

Already in 1946, the urgency for firing tables ebbing away with the end of the war, scientists from Los Alamos came to visit the ENIAC for as yet classified computations to check the feasibility of making a hydrogen bomb. Edward Teller had conceived the idea of a cascade reaction of neutrons and had made a design based on this simple idea. The problem, however, was to check and test this basic idea and get a view on the destructive impact of such a scheme.

> To get the recipe just right, the Los Alamos scientists had to calculate what was happening inside the reaction at increments of one ten-millionths of a second. They had made some crude calculations with slide rules and educated estimates, and mathematician Stanislaw Ulam had begun to raise doubts about Teller's design. (McCartney, 1999, 103)

Though Nick Metropolis and Stan Frankel sighted the ENIAC in 1946, it was only later, 1948 onwards, that computations were really executed. In the mean while the programming of the ENIAC was completely revolutionised.

2 Function Tables turn into Instruction Tables

> The ENIAC was a son-of-a-bitch to program. The number of function switches was so limited, programs had to be broken up into rather small pieces (Jean Bartik, Interview 1996)

To solve the problem that programming the ENIAC took so much time, the stored-program concept was developed. P. Eckert seems to have come up with the idea in a memo 1944, but it was John von Neumann who published the idea in his *First Draft* (von Neumann, 1945), capering away yet another idea of Eckert and Mauchly's. Though the idea was conceived to be implemented in the next generation computer, it was implemented on the ENIAC 1947-1948 using the 'function table panel'. Most probably, the intricacy of the H-Bomb equations (see infra) triggered the actual reorganization of the ENIAC programming schemes.

> In the meantime [while Metropolis and others prepared problem set-ups for the H-Bomb] Richard Clippinger, a staff member at Aberdeen, had suggested that the ENIAC had sufficient flexibility to permit its controls to be reorganized into a more convenient (albeit static) stored-program mode of operation. This mode would have a capacity of 1800 instructions from a vocabulary of about 60 arithmetical and logical operations. The previous method might be likened to a gigant plugboard, that is to say, to a can of worms. Although implementing the new approach is an interesting story, suffice it to say that Johny's wife, Klari, and I designed the new controls in about two months and completed the implementation in a fortnight. (Metropolis, 1987, 128)

The design was on paper in 1947, when the first meeting of the Association for Computing Machinery (ACM) was held on the Aberdeen Proving Ground, and implemented and tested beginning of 1948.
R.F. Clippinger (1948) described the converter code that was used in the table lookup scheme, a more transparent presentation can be found in (Neukom, 2006). Luckily enough for those wiring this converter code, the engineers of the ENIAC had already provided for something like a lookup table.

> The idea was to encode the instructions of a problem on the "function tables," three panels of the machine, each of which bore 1,200 ten-way switches. They had been intended as a computer-accessible table lookup, e.g. for empirical functions, but it was now proposed to use them to set up the succession of instructions, each represented by a two-digit number. The wiring of the plug boards would be set up permanently on the machine in a way that would cause the machine to read a number from the table, carry out the instructions encoded by it, fo on to reading the next number, etc. Thus, the background wiring played the role of a present-day "compiler" – more specifically, of an interpretative routine, since the source code had to be read and interpreted anew for each run, and no permanent object code was set up. This mode of operation would slow down the machine, of course; it was estimated that its speed would decrease at least by a factor of 5, a small price to pay for eliminating the long set-up time. (Alt, 1972, 694)

The idea of using a table to shortcut calculations – or in this case to shortcut programming the calculations, since the calculations took more time – is, however, hardly new in some way, and even connects most smoothly to the history of mathematical tables.

> There are numbers, proportions, formulae and calculations that deserve to be done and written down once and for all, because they occur very often, so as to avoid the trouble to find or calculate them over and over again. This is the reason why in all parts of mathematics one has tried to put everything in tables that can possibly be put in tables. (Lambert, 1770, 1)[7]

Perfecting this idea, however, has the queer effect that everything behind the table becomes invisible: A person using logarithmic tables doesn't know anymore how to produce such a table, a person programming with a converter code gets oblivious of the wiring behind it. At least one early user of the ENIAC regretted this re-wiring. In D.H. Lehmer's opinion, "von Neumann spoiled the ENIAC" (Lehmer, 1976) and in a later statement Lehmer foresees but too clearly the problem that the software will one day obscure the hardware:

> By 'software' is meant the system of executive programs used by the machine operators and programmers to put jobs through the machine. [...] This laudable purpose is important for the businesslike operation of a computer for an industry or a university that behaves like an industry but imposes restraints on any particular user, especially a low priority user like the number theorist. (Lehmer, 1969, 120)

Anyway, it is with the advent of the alphabet of elemental instructions for a computers that the work of generations of logicians enters the computer world.

> [Von Neumann] discussed the need for, and likely impact of, electronic computing. He mentioned the "new programming method" for ENIAC and explained that its seemingly small vocabulary was in fact ample: that future computers, then in the design stage, would get along on a dozen instruction types, and this was known to be adequate for expressing all of mathematics. (Alt, 1972, 694)

Indeed, many logicians – as von Neumann himself once – had 1860–1930 come up with various schemes to formalize and reduce the fundamental building blocks of logics and mathematics. A part of Hilbert's programme went in this direction, Russell and Whitehead's *Principia*, Skolem's, Schönfinkel's and Gödel combinatory and resymbolizing approaches, Emil Post's most singular but brilliant study of formal systems &c. And of course, von Neumann had read Turing's paper on computable numbers, that shows that a universal 'machine' (with infinite tape) can be constructed with only a limited number of symbols. It is to this idea and its whole tradition and history von Neumann alludes. Indeed, in the subsequent evolution, much changed to the instruction tables and early 'compilers'. In the late 1940ies, however, programming was still more something like a lookup table.

3 Sequential Programming (1948–1955)

3.1 Airflow Problem

In his paper on the converter code, Clippinger gives an example program, but is seems somewhat garbled (Neukom, 2006, 1). However, he also describes in *MTAC* another problem that most likely served as a more complicated test program early 1948.

> Given a body of revolution with sharp nose immersed in a steady uniform flow at sufficiently high Mach number, it is permissible to neglect viscosity and body forces.
>
> If the independent variables α and β, constant on characteristics, are introduced, it can be shown that the differential equations take the form
>
> $$\begin{aligned} H_{y_\alpha} - (K+R)x_a &= 0 \\ H_{y_\beta} - (K+R)x_b &= 0 \\ H_{u_\alpha} + (K-R)v_\alpha + (P+Q)x_\alpha &= 0 \\ H_{u_\beta} + (K+R)v_\beta + (P-Q)x_\beta &= 0 \\ dz &= yA(-vdx+udy) \end{aligned}$$
>
> where x and y are cylindrical coordinates of a point P, u and v are the corresponding velocity components, and H, K, R, P, Q, A are known function of y, u, v, z.
> The boundary conditions which complete the formulation of the problem are of the form
>
> $$dy/dx = F(x,y) \quad v/u = F(x,y)$$
>
> on the contour of the body, and
>
> $$G(u,v) = 0 \quad dx/dy = H(u,v)$$
>
> on the shock-wave. (R.E. Clippinger, *MTAC* (1948) 3 (23), 206)

All this leads to a set of 5 differential equations, conditioning a function with 9 parameters. The accuracy desired is a second order approximation for this function. The physical problem encrypted in these equations is testing a certain airplane model under extreme circumstances. The second order approximation obtained must have been disappointing, not really converging to a real situation, since von Neumann is renowned to have said that it might be better for certain problems such as airflow problems to construct a wind tunnel testing the plane, to get at the solutions of the differential equations – thereby reducing the mathematics back to physics.

3.2 Classified (continued) and New Methods

Coming back to the classified problem, the set-up for tracking down the feasibility of the H-Bomb had proven 1947–48 to be a complex and tough problem. The original idea still seems to have been Teller's neutron cascade from 1944, but in the mean time Ulam and Everett – Ulam spawning the ideas and Everett handling his slide rulers – had proven Teller's original model and calculations probably wrong, if not just sloppy. The feasibility of the H-Bomb was again an open question, the main problem being the first microseconds of the fission part of the H-Bomb, that had to start the fusion process. Initially (until 1946), it was thought that a fission would rather unproblematically ignite the fusion process, but Ulam and Everett's calculations showed that a hydrodynamic effect (the Compton effect) cooled

down the process. Several other schemes were proposed by Teller (1947–8), but eventually it was Ulam who came up with the idea to use the radiation effect of the fission (1948–9). In a still classified paper by Ulam and Teller, a scheme is explained how to 'bundle' radiation and use it as a 'lense' to 'mirror' the neutrons, in short, some very controlled geometrization of the effects after fission to compress the particles for fusion. This explains why Los Alamos scientists published papers in public journals on their theoretic advances and numerical methods, of course describing a very general set-up of the problem, that stand in a relationship to the neutron and the hydrodynamic part of the Bomb, but that not a trace of work on the radiation ever appeared.[8]
The neutron part of the problem is something like this:

> 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.
> At each stage a sequence of decisions has to be made based on statistical probabilities appropriate to the physical and geometric factors. The first two decisions occur at time $t = Q$ when a neutron is selected to have a certain velocity and a certain spatial position. The next decisions are the position of the first collision and the nature of that collision. If it is determined that a fission occurs, the number of emerging neutrons is eventually followed in the same fashion as the first. If the collision is decreed to be a scattering, appropriate statistics are invoked to determine the new momentums of the neutron. When the neutron crosses a material boundary, the parameters and characteristics of the new medium are taken into account. Thus, a genealogical history of an individual neutron is developed. (Metropolis, 1987, 127)

It is this complex if not chaotic neutron interaction scheme[9] that prompted the invention of the Monte Carlo method, the original idea stemming from Ulam, the development coming from von Neumann and, somewhat later, Metropolis and others, though the method had been independently anticipated by Enrico Fermi (Metropolis, 1987, 128) (Doolen and Hendricks, 1987, 142).
Also, the Los Alamos scientists succeeded in splitting up the problem in parts of increasing complexity. The complex problem was thus divided into three main subproblems: Position and velocity of the neutrons, the hydrodynamic effects and the radiation. Only the first and second problem were, at least in a first stage, calculated on the ENIAC, the radiation tests were done on the MANIAC, that was especially designed for nuclear riddles. Monte Carlo methods, however, were used for all of them:

[8] General papers on radiation, for use in nuclear reactions, do appear, but later in the 1960ies.
[9] For the lovers of equations:
A very simplified version of such a problem would lead to the equation: $\frac{\partial u(x,y,z)}{\partial t} = a(x,y,z)\Delta u + b(x,y,z)u(x,y,z)$ where $u(x,y,z)$ represent the density of the particles at the point x, y, z. The Laplacian term, $a\Delta u$ on the right hand side corresponds to the diffusion of the particles, and bu to the particles procreation, or multiplication.(Metropolis and Ulam, 1949, 338)
A little more specifically, Schrödinger equations come in:
For example, as suggested by Fermi, the time-independent Schrödinger equation $\Delta\psi(x,y,z) = (E - V)\psi(x,y,z)$ could be studied as follows. Re-introduce time dependence by considering $u(x,y,z,t) = \psi(x,y,z)e^{Rt}$ u will obey the equation $\frac{\partial u}{\partial t} = \Delta u - Vu$ This last equation can be interpreted however as describing the behavior of a system of particles each of which performs a random walk, i.e., diffuses isotropically and at the same time is subject to multiplication, which is determined by the value of the point function V.(Metropolis and Ulam, 1949, 340–41)

> for example, a neutron-velocity spectrum with various peaks and valleys is difficult to handle mathematically. For Monte Carlo one needs only to mirror the velocity spectrum in the probability distribution. Also, the Monte Carlo method is sufficiently flexible to account for hydrodynamic effects in a self-consistent way. In an even more elaborate code, radiation effects can be dealt with by following the photons and their interactions. (Metropolis, 1987, 127)

The increase in complexity can be easily understood if one takes the fact into account that these three layers of the problem are ordered in decreasing magnitude of intra-particular forces, and at the same time in increasing magnitude of interaction distance. In other words, from a collection of balls to a fluid to something gas-like, the statistical methods becoming trickier due to the smaller coefficients, larger variables and extracurricular effects.
As Ulam calls it, it is a "combination of stochastic and deterministic flows" (Metropolis and Ulam, 1949, 341), for which neither analytic nor statistical methods alone are satisfying. This wasteland in between tractable deterministic systems and complex though uniformly behaving systems, methodically between algebra/analysis and statistics, had to be colonized to track down the feasibility of the H-Bomb. The beginning of branching processes and chaos theory (Ulam and Everett, 1948), of cellular automata (Pasta and Ulam, 1959, original report from 1952), and of the Monte Carlo method in its various forms are all attempts at colonizing.
[A longer digression on the details would be nice, certainly since quite a lot of materials have become declassified in the late 1990ies, and can be found at http://fas.org/sgp/othergov/doe/lanl/.]
[Also: The at least two variants of the Monte Carlo method – the original one in (Metropolis and Ulam, 1949), that ran on the ENIAC, and the one modified for tracking radiation in (Metropolis et al., 1953), that ran on the MANIAC – deserve closer attention, especially the tricky point of random number generators. It seems that all Los Alamos scientists stuck to von Neumann's middle square method, although they realized its insufficiency. Von Neumann is reported to have said it did not matter really, and nobody apparently contradicted him. Already 1949, however, Lehmer (1951 (1949) had proposed the much better linear congruential method, so the attachment to the middle square is a bit strange. Again, von Neumann is reported to have said, the middle square method had the advantage of being faster than other methods, which seems very unlikely given the simplicity of linear congruential generators. Methinks, some strange twisted subhistory lies hidden here. Its bottomline might be that von Neumann might figure twice in the bizarre Brewer's dictionary of Phrase and Fable*, where under the lemma* invention *no explanation of invention is given, but a list of inventors who died of their own invention. Twice in the sense that von Neumann died of the radiation effects of the H-bomb test (1), and perhaps that the radiation calculations were off the mark because of a bad random number generator (2)]*

3.3 Slow Moses

The first 'background process' was also introduced on the ENIAC. When nothing was computed on the ENIAC, "Slow Moses" got some run time. George Reitwiesner installed this background program, that was looking for a Fermat quotient q_2, after Lehmer's sieve, the second number theoretic problem run a digital computer.

> The problem here was to find for each prime p to so-called Fermat quotient p_2 defined by
>
> $$2^{p-1} - 1 \equiv pq_2 \pmod{p^2}$$
>
> Particular interest is attached to those p for which $q_2 = 0$ for only when this happens can there be a solution of Fermat's equation $X^p + Y^p = Z^p$ in integers X, Y, Z prime to p. (Lehmer, 1974, 5)

However, due to the fact that "the ENIAC was drastically altered from the most parallel computer ever built to a slower serial machine somewhat like the one bottleneck machines of today", and the fact the programming of the problem was not optimized at all, only value $p = 25000$ was reached after several years. No $q_2 = 0$ were found, quite logically as we know anno 2006, since Fermat's problem has no solutions for $p > 2$.

3.4 Doppler V-2

1949 the first comparative computer test was conducted – what computing machinery is the best bargain? The intended custumor are of course the U.S. Army services, the test problem consequently close to their interests: Following the trajectories of a V-2 rocket. In the running, the following machinery: Standard IBM equipment; IBM Relay Multipliers; the ENIAC; the Bell Labs Computing Machine. That is, 3 electro-mechanical relay computers against one electronic monster with vacuum tube relays.

> DOVAP (Doppler Velocity and Position) is a radio-Doppler method for the determination of the coordinates of the trajectories of long-range V-2 rockets launched at the Army's White Sands Proving Ground, New Mexico. [...] In this system, continuous-wave radio signals are sent from a transmitter to a transceiver in the missile and to each of several ground station receivers. In the missile-transceiver the signals are modified by a frequency-doubling operation and then are retransmitted to each of the several ground stations. In the ground receivers the frequency of the signals received directly from the transmitter is likewise doubled, and these double-frequency signals are mixed with those received from the missile. The mixed signals are recorded on 35 mm movie film simultaneously for the several receivers at one master receiver station. (Hoffleit, 1949, 373)

This near synaesthetic coalition of technology enables a count of Doppler shifts. That is, a wave is transmitted from one source, and two wave-lengths are received at two different points, due to different distances and the velocity, these two wave lengths will differ slightly, this is the Doppler effect. The time pulse with which the waves are recorded is the unit measure for counting the magnitude of the Doppler wave length shift.
Assuming the position of the rocket launch known relative to the receiver posts (u_i), the data recorded at a post i at a given time are $(u_i)_n = c_i$, the Doppler wave length unit is λ and the count at a station i is N_i than the following relation holds:

$$u_i = c_i + N_i\lambda$$

The problem then is to calculate the u_i (having three coordinates x, y, z) at time t different from launching time. If there are three receiver stations, these u_i can be unambiguously found. Imagine three spheroids with a receiver station as a center, and radius found through the above formula, these three spheriods meet in one point only that is not underground, this is the desired solution. Thus, the equation is the difference of two spheres:

$\sqrt{x^2+y^2+z^2}+\sqrt{(x_i-x)^2+(y_i-y)^2+(z_i-z)^2}=u_i$ (i=1, 2, 3)

This then is succesively approximated through finite differences.
Implementing this problem on the four computing machines, only the ENIAC comes close to real-time computation,

> Barring machine failures (including failures due to warped cards or cards affected by humidity or static charges) the actual computation on the ENIAC require little more time than the time of flight of the missile. (Hoffleit, 1949, 376)

To be precise, in man hours the problem for 800 points at half-second intervals on the trajectory of a 400-second time-of-flight missile would take 10 weeks. In the test, the standard IBM equipment would practically take near 10 weeks, the relay calculators 2 weeks, the ENIAC in real time (400 seconds) were it not for the 2 programming days needed, and the Bell machines 70 hours. Conclusion of the test

> It appears from the test [...] that the complete DOVAP computations for a single 800-point trajectory are done equally expeditiously on the ENIAC or the Bell machines. [...] because of the high programming time of the ENIAC, however, the Bell machines are to be preferred for shorter problems like the computations for a few points on a single trajectory. For longer problems the picture changes. For example, the computations on 10 trajectories for which the raw data are available simultaneously would be completed in about $2\frac{1}{2}$ days on the ENIAC, [...] but would require almost 30 days of continuous day and night running time on the Bell machines. (Hoffleit, 1949, 377)

3.5 Squaring the Circle: π

Exactly 4 years after the first long run test, on yet another Fourth of July weekend – 1950 in this case – the ENIAC did not have to use its circuitry on yet another differential equation problem, but could for the first time spare some time and pursue leisure-related interests, the computation of digits of π and e. Again it is John von Neumann who brings up the idea, and someone else, G.W. Reitwiesner, carries it out.

> 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. (Reitwiesner, 1950, 11)

The test with e was executed first, "to gain expertise and technique for the more difficult and longer project on π", using the formula:

$$e=\sum_{n=0}^{\infty}(n!)^{-1}$$

For π many (arctangent) formulae were at disposal, but the formula with the easiest denominators (because of the accuracy of division on the ENIAC) was chosen:

$$\pi/4=4\quad\arctan\quad 1/5-\quad\arctan\quad 1/239\quad\text{with}\quad\arctan\quad x=\sum_{n=0}^{\infty}(-1)^n(2n+1)^{-1}x^{2n+1}$$

e was calculated on a Fourth of July weekend, π an a Labor Day weekend.

Since many digits of e and π were desired, the size of the storage rings (10 or 20 digits if coupled) had to be circumvented. Therefore the division process was split up: A number i of digits to be calculated per run was determined, the i digits found by division and the remainders were stored separately, these last ones printed out on punch cards. After all first i digits of all terms in the formulae for e and π were calculated and added up (and stored on another punch card), the process was started again, but this time starting from the remainders on the punch cards, to get at the next i digits &c. At last, all digit punch cards were added (in the case of e), or all term-wise digit cards were added, and then the terms in their turn added (for π). As far as regards the sequence of commands, these were quite straightforward with the new converter code. The formula for e uses only multiplication, division and addition, and is thus easy to program, certainly with the lookup table of elementary operations, π is different, since the *arctan* implies powers n, -1 and $2n+1$ and is thus a little more intricate. Anyway, the splitting in chunks of digits and remainders is most of the work. The best-known result of this ENIAC-computation is that the distribution of the digits of π obey a normal distribution, and are thus 'random', whereas the digits of e diverge from a normal distribution, and are not random (Metropolis et al., 1950), a result confirmed by later computations.

Some annotations on the efficiency of the ENIAC π calculation should be made. Machin's formula for π, which von Neumann proposed to Reitwiesner, is a classic for π computations, because it involves but two terms and rather small coefficients and is thus quite suited for numerical or mechanical computation. Though Machin's formula has a long tradition, at least since 1938 it was known how to construct better formulae of the same kind. Apparently while studying parts of C.F. Gauss's *Nachlass*, J.W. Wrench Jr. and D.H. Lehmer happened upon the 'Cyclometric Tables', that contain several arccotangent formulae for π. In (Wrench, 1938) and (Lehmer, 1938) they develop the theory implicit in Gauss's tables, using some results by Størmer.[10]

Although the arc(co)tangent is a transcendental function, its theory is strongly related to number theory, because the arctangent's derivative is $\frac{1}{1+n^2}$. Factorizations of numbers of the form n^2+1, instances of the so-called binomial factorizations, are of a certain interest in number theory.

> The problem of expressing a rational multiple of π as a sum of arccotangents, however, can be solved in an infinite number of ways, most of them uninteresting. While I do not presume to set down any hard and fast rules for this indoor sport, still I should like to point out certain possibilities as well as certain inescapable facts which should not be overlooked by those seeking interesting new relations for π, and to give a rough scheme for comparing such relations which each other. (Lehmer, 1938, 657)

Lehmer indeed defines a rough measure for the efficiency of such formulae. In this scheme, Machin's formula has measure 1.8511, but Lehmer lists 11 formulae having lower measures. In particular, a formula from Klingenstierna with measure 1.2892 is given, that takes advantage of shortcuts possible in the decimal digit system[11], and thus 50 percent more efficient than the one used by von Neumann/Reitwiesner. Actually, though von Neumann

[10]The rediscovery of the 'computational Gauss' by the American number theorists J.W. Wrench, D.H. Lehmer and D. Shanks, in parallel with the evolution of applicating digital machinery to number theory is in itself a worthwhile chapter which we intend to present some other time.

[11]Lehmer makes a conservative estimate on $arccotan 10 = arctan \frac{1}{10}$, giving it measure 0.5, probably, 0.25 would be closer to the truth, giving a measure 1.0392, cfr. the time estimate in (Shanks and Wrench, 1962, 77).

had listed Klingenstierna's formula as a possible candidate, Reitwiesner had rejected it, and used Machin's formula so that "identical programming apply for all series employed" (Reitwiesner, 1950, 13). Commodity of programming apparently excludes shortcuts.
Only in 1962 Daniel Shanks and J.W. Wrench would compute π to 100,000 digits on an IBM 7090, using a new arccotangent identity, though not the one Lehmer recommended, but a newly derived identity. Klingenstierna's formula was very suited to a decimal system, but since in the 1960ies nearly all computers used the binary system, other formulae had to be found. Exactly such a formula is used, with a measure of 2.0973 in a decimal computing machine, but with a measure of only 1.2400 on a binary machine, where $\frac{1}{3}$ of the divisions can be shortcut as binary shifts. Shank and Wrench's article ends beautifully with 20 pages of tiny printed digits of π (Shanks and Wrench, 1962, 80-99), probably one of the last times such an extent of π-digits were put on paper.

References

(1948/1950), *Proceedings of the first symposium on Large-Scale Digital Calculating Machinery*, Harvard University Press, Cambridge (MA).

(1951 (1949)), *Proceedings of a second symposium on Large-Scale Digital Calculating Machinery*, Harvard University Press, Cambridge (MA).

Alt, F. (1972), "Archaeology of Computers – Reminiscences, 1945–1947", *Communication of the ACM*, 15 (7), pp. 693–694.

Bullynck, M. (2005), "Decimal Periods and their Tables: A Research Topic (1765-1801)", (forthcoming).

Clippinger, R. (1948), *A Logical Coding System Applied to the ENIAC*, BRL 673, Ballistic Research Laboratories, Aberdeen Proving Ground.

Courant, R. (1948/1950), "Method of finite differences for the solution of partial differential equations", In (Lar, 1948/1950).

Dederick, L. (1940), "The Mathematics of Exterior Ballistic Computations", *American Mathematical Monthly*, 47 (9), pp. 628–634.

Doolen, G. and Hendricks, J. (1987), "Monte Carlo at Work", *Los Alamos Science*, Special Issue, pp. 142–143.

Goldstine, H. and Goldstine, A. (1946), "The Electronic Numerical Integrator and Computer (ENIAC)", *Mathematical Tabels and Other Aids to Computation*, 2 (15), pp. 97–110.

Hoffleit, D. (1949), "A Comparison of Various Computing Machines Used in the Reduction of Doppler Observations", *Mathematical Tabels and Other Aids to Computation*, 3 (25), pp. 373–377.

Howlett, J.; Metropolis, N. and Rota, G.-C. (eds.) (1980), *A History of Computing in the Twentieth Century*, Academia Press, New York.

Lambert, J. (1770), *Zusätze zu den Logarithmischen und Trigonometrischen Tabellen*, Spener'sche Buchhandlung, Berlin.

Lambert, J. H. (1765/1767), “Mémoire sur la resistance des fluides avec la solution du probleme ballistique”, *Histoire de l’Académie Royale des Sciences et Belles-Lettres de Berlin*, pp. 102–188.

Lasalle, J. (ed.) (1974), *The influence of computing on mathematical research and education*, vol. 20 of *Proceedings of Symposia in Applied Mathematics*, American Mathematical Society.

Lehmer, D. (1933a), “Numerical Notations and Their Influence on Mathematics”, *Mathematics News Letter*, 7 (6), pp. 8–12.

— (1933b), “A Photo-Electric Number Sieve”, *American Mathematical Monthly*, 40 (7), pp. 401–406.

— (1938), “On Arccotangent Relations for π”, *American Mathematical Monthly*, 45 (10), pp. 657–664.

— (1951 (1949)), “Mathematical Methods in Large-Scale Computing Units”, In: (Lar, 1951 (1949, 141–146).

— (1953), “The Sieve Problem for All-Purpose Computers”, *Mathematical Tabels and Other Aids to Computation*, 7 (41), pp. 6–14.

— (1969), “Computer Technology applied to the Theory of Numbers”, In: (LeVeque, 1969, 117–151).

— (1974), “The influence of computing on mathematical research and education”, In (Lasalle, 1974, 3–12).

— (1976), “A history of the sieve process”, In: Howlett et al. 1980, 445–456.

LeVeque, W. (ed.) (1969), *Studies in Number Theory*, vol. 6 of *Studies in Mathematics*, Mathematical Association of America, distributed by Prentice-Hall.

McCartney, S. (1999), *ENIAC. The Triumphs and Tragedies of the World’s first Computer*, Walker and Co., New York.

Metropolis, N. (1987), “The Beginning of the Monte Carlo Method”, *Los Alamos Science*, Special Issue, pp. 125–130.

Metropolis, N.; Reitwiesner, G. and von Neumann, J. (1950), “Statistical Treatment of Values of First 2,000 Decimal Digits of e and of π Calculated on the ENIAC”, *Mathematical Tabels and Other Aids to Computation*, 4 (30), pp. 109–111.

Metropolis, N.; Rosenbluth, A.; rosenbluth, M.; Teller, A. and Teller, E. (1953), “Equation of State Calculations by Fast Computing Machines”, *Journal of Chemical Physics*, 21 (6), pp. 1087–1092.

Metropolis, N. and Ulam, S. (1949), “The Monte Carlo Method”, *Journal of the American Statistical Association*, 44, pp. 335–341.

Neukom, H. (2006), “The second life of ENIAC: ENIAC’s Converter Code”, *Annals of the History of Computing IEEE*, (Web Extra).

von Neumann, J. (1945), *First Draft of a Report on the EDVAC*, Tech. rep., Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia.

Pasta, J. and Ulam, S. (1959), “Heuristic Numerical Work in Some Problems of Hydrodynamics”, *Mathematical Tabels and Other Aids to Computation*, 13 (65), pp. 1–12.

Pell, J. (1672), *Tabula Numerorum Quadratorum Decies millium, una cum ipsorum Lateribus ab Unitate incipientibus ordine naturali usque as 10000 progredientibus*, Printed by Thomas Ratcliffe and Nath. Thompson, and are to be sold by Moses Pitt at the White Hart in Little Britain, London.

Reitwiesner, G. (1950), “An ENIAC Determination of π and e to more than 2000 Decimal Places”, *Mathematical Tabels and Other Aids to Computation*, 4 (29), pp. 11–15.

Shanks, D. and Wrench, J. W. (1962), “Calculation of π to 100,000 Decimals”, *Mathematics of Computation*, 16 (77), pp. 76–99.

Szabó, I. (1979), *Geschichte der mechanischen Prinzipien und ihrer wichtigsten Anwendungen*, Basel.

Ulam, S. and Everett, C. (1948), *Multiplicative Systems in Several Variables I, II, III*, Tech. Rep. LA-683; LA-690; LA-707, Los Alamos Laboratory.

Wrench, J. (1938), “On the Derivation of Arctangent Equalities”, *American Mathematical Monthly*, 45 (2), pp. 108–109.