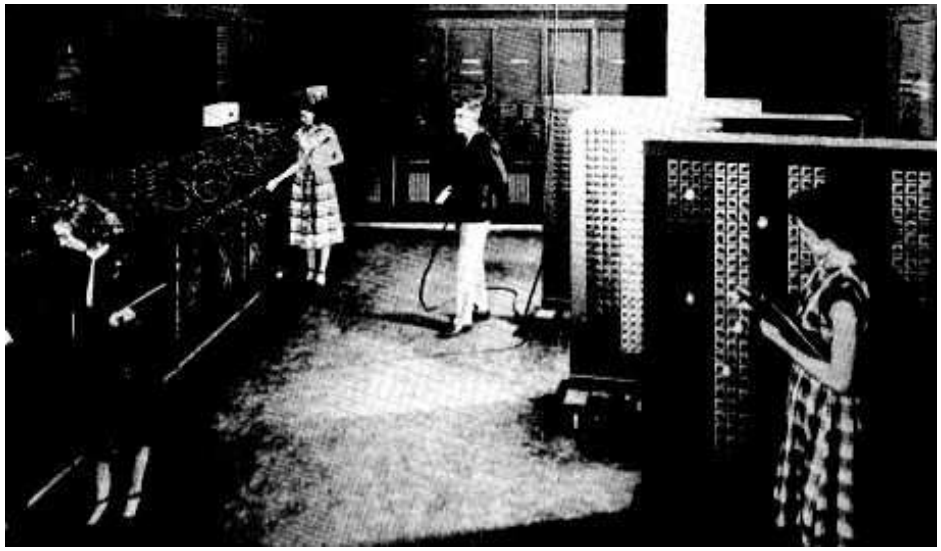
ENIAC, matrix of numerical simulation(s!)



M. Bullynck[1], L. De Mol[2], and M. Carlé[3]

[1] Paris 8, maarten.bullynck@kuttaka.org

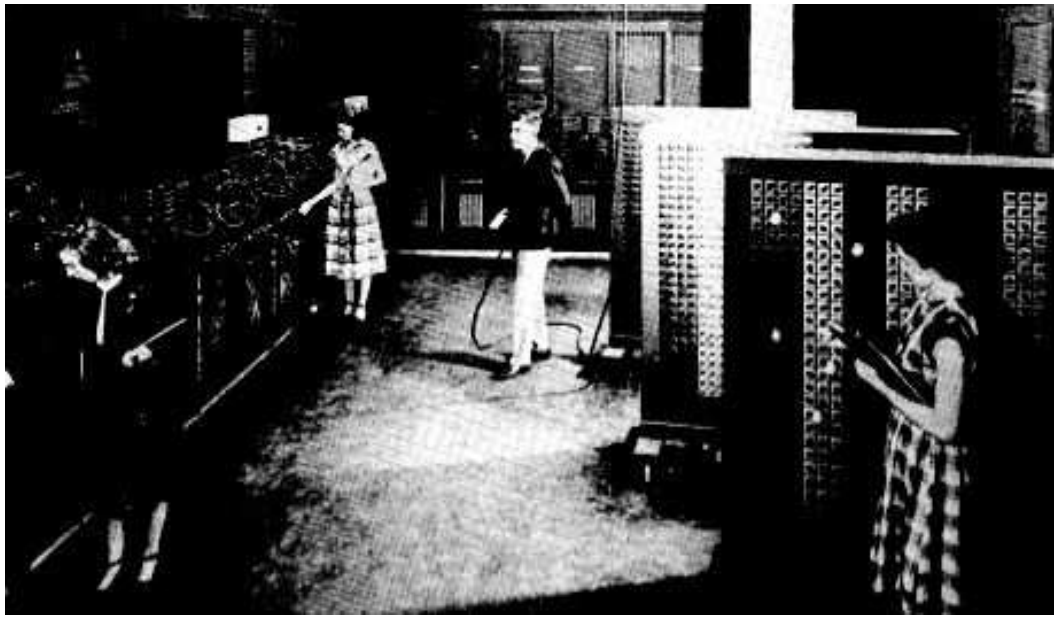
[2]Universiteit Gent, elizabeth.demol@ugent.be

[3] Athens, mc@aiguphonie.com

Dedicated to the memory of Friedrich A. Kittler (1943–2011)

ENIAC, Grundgefüge numerischer Simulationen

oder:
Es gibt keine Simulation



M. Bullynck, L. De Mol, and M. Carlé

Introduction: Engaging with the ENIAC, engaging with computing

General motivation ENIAC as really the first electronic , digital and programmable machine – a historical discontinuity → its impact often underestimated in the literature

Our approach A techno-historical one: Digging into the details of this machine and the interaction*s* with it (see e.g. De Mol, Bullynck, Carlé, 2010 on Curry, De Mol and Bullynck 2008 and 2010 on Lehmer)

ENIAC as matrix of simulation Simulation was basically invented for and because of ENIAC.

Introduction: Engaging with the ENIAC, engaging with computing

ENIAC as matrix of simulation Simulation was basically invented for and because of ENIAC.
We are not interested in simulation as test for a model, but in the **the matrix of interrelations between man and ENIAC (machine), embedding, enabling and shaping simulation**.
This happens on a **threefold level**

1. the mathematics (numerical methods);
2. the logical organisation of the program (translation into a computer program);
3. the physicality of the computer.

Why we claim (tongue in cheek): 'There is no simulation'

Historical reasons: **Before the ENIAC, there was no numerical simulation**, the sole idea of simulation only became possible because of:

- The thousandfold speed up provided by ENIAC (and successors)
- The programmability of ENIAC

Epistemological reasons: We believe that simulation need not have an epistemological status in its own right, because, in our approach, it is **one of the effects of man-machine interaction**

The role of numerical methods on ENIAC: Numerical methods were developed because of ENIAC's limits and possibilities, as a **medium that sizes and transmutes the operator's view on a problem to the machine and vice versa**, rather than as "a necessary medium between the theoretical model and the simulation"

Meet the ENIAC

Meet the ENIAC

- First **general-purpose electronic digital computer** worldwide, speed of 5000 additions per second. Publicly presented in 1946, and only (public) specimen of its kind until 1949! About 100 differents sorts of computations were run on ENIAC (according to the list by Barkley Fritz)
- 1946–1947/1948: Original set-up, a **modular and parallel** machine with external programming by cables.
  Programming the ENIAC in its original configuration thus came down to *"the design and development of a special-purpose computer out of ENIAC component parts"* (B. Fritz); or the ENIAC *"was a son-of-a-bitch to program"* (Jean Bartik)
- Working memory confined to 20 words (=accumulator); Constants stored in constant transmitter or function tables; New information could be fed to the ENIAC by punched cards

On the ENIAC, there are **Trade-off**s between:

- **logical complexity** of a program and the **amount of memory** available: an accumulator is used either for working memory, or for doing discriminations
- **computational speed** and **number of data used**: computation is fast (5000 additions per second), reading or punchung results is slow (.3 or .4 second)

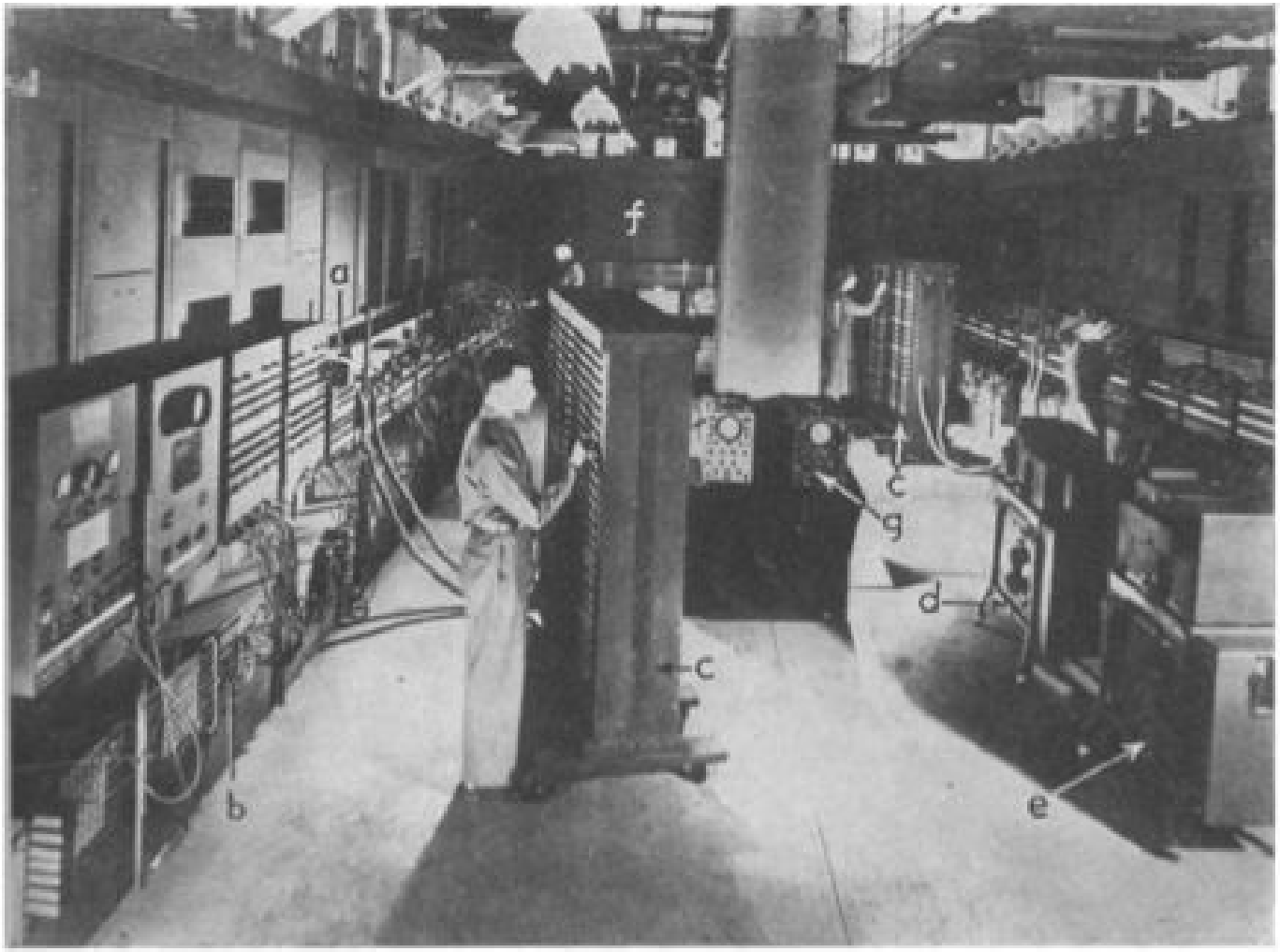


Fig. 1. General view of the ENIAC. *a*, digit trays (see fig. 2); *b*, program trays (see fig. 2); *c*, function unit (see fig. 2); *d*, card reader; *e*, card punch; *f*, high-speed multiplier; *g*, testing equipment

View inside the ENIAC

ENIAC 'sparks' new numerical methods

Rethinking numerical methods

Since on the ENIAC **computation is 'cheap'**, but **set-up of a program and memory 'expensive'**, simplicity of algorithms is important.

- Classes of numerical methods for hand and desk calculators (such as explicit long formulae) are not suited (need too much memory)
- Classes of other numerical methods could now be implemented with success (that take lots of computation)
  - Iterative methods
  - Parallel methods
  - Number crunching methods, e.g. of a mixed deterministic and stochastic nature

Two examples of newly developed numerical methods on ENIAC:

- Schoenberg and Curry: **Splines** (1946)
- Ulam and von Neumann: **Monte Carlo method** (1947)

Question: **How are these numerical methods used, developed and coded?**

And what is their impact on all three levels?

Splines

"The advantage of an iterative process are that it is eminently suitable for the Eniac" (Curry and Wyatt)
Iterative algorithms work well (are convergent) for the main function of the ballistic problem, but adding the secondary functions (drag, resistance) poses a challenge

- Since ENIAC's memory is at a premium, one has to choose between: 1) **Simplifying** the main scheme, liberating some accumulators to add secondary functions; or: 2) **combining cycles of computation**: *"run with the basic scheme first, and then use the output cards of this run as primary cards for a new run [...] composite interpolation, primary cards give t=t(x), secondary cards y=y(t), output is y=y(tx)) "*
- The secondary, empirical functions are only roughly tabulated, which works for explicit calculation with desk calculators, but **accumulates errors** when used in an iterative procedure: *"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."* (Schoenberg)

Solution: **Splines**, instead of one interpolation function/polynomial, use a **'broken' polynomial to smooth the rough data**

Monte Carlo

"[Ulam] realized that with such increased computing power it was appropriate to revive model- or statistical sampling techniques."

- Presentation of results by ENIAC on a model for the thermonuclear device Super by Metropolis and Fraenkel, gets Ulam thinking about
  1. **speed** of electronic devices;
  2. statistical **sampling techniques** that could now be done fast on a large scale computer (and used in neutron diffusion calculations)

  further developed by von Neumann, Richtmyer etc.
- **Mixture of deterministic and stochastic processes**
  " 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. [...] Thus, a genealogical history of an individual neutron is developed. The process is repeated for other neutrons until a statistically valid picture is generated. " (Metropolis)

Engaging with the ENIAC

Machine reflections on the numerical procedures: Coding

Coding the numerical procedures: Both Splines and Monte Carlo use **intricate cascades of discriminations** (branching) on the ENIAC

- Splines: to decide on **how to break up the polynomial or the numerical procedure** and at what places
- Monte Carlo: to model the **'decision tree'** of a particle

Coding of the numerical procedure:

- Curry: **to adapt it to the ENIAC** and adding error routines to **feed back to the operator**
- The Monte Carlo people: **to jam the ENIAC full with data** and follow the **development of its 'meaning'**

Engaging with the ENIAC: Von Neumann versus Curry in a nutshell I

- Von Neumann: *"The computational execution would be something like this: Each neutron is represented by a card C which carries its characteristics i, r, s, v, t, and also the necessary random values A, p, v, p, p', p", p"' [...] the instructions given on this 'computing sheet' do not exceed the 'logical' capacity of the ENIAC. I doubt that the processing of 100 'neutrons' will take much longer than the reading punching, and (once) sorting time of 100 cards, i.e., about 3 minutes. Hence, taking 100 'neutrons' through 100 of these stages should take about 300 minutes, i.e., 5 hours."*
  Basic unit in von Neumann's thinking **a card= a particle in the (mathematical) model**; most of the time is spent on collating cards

- Curry and Wyatt: *"The stages can be programmed as independent units, with a uniform notation as to program lines, and then put together; and since each stage uses only a relatively small amount of the equipment the programming can be done on sheets of paper of ordinary size."*; every stage is an "autonomous piece of computation"
  Basic unit in Curry's thinking, **one preparatory programming sheet = an elementary program (function)**; main question is how to combine programs to save either memory or time

Engaging with the ENIAC: Von Neumann versus Curry in a nutshell II

- Von Neumann: *"A common set-up of the ENIAC will do for all criticality problems. In changing over from one problem of this category to another one, only a few numerical constants will have to be set anew on one of the 'function table' organs of the ENIAC."*
  Program does not change, only its constants;
  main problem is finding an algorithm for a given problem, not programming the algorithm

- Curry and Wyatt: *"The problem of program composition was a major consideration in a study of inverse interpolation on the ENIAC [...]; for although that study was made under stress and was directed primarily towards finding at least one practical method of programming a specific problem, yet an effort was made to construct the program by piecing together subprograms in such a way that modifications could be introduced by changing these subprograms."*
  Provision to extend and locally change program and add variables, or **trade in logical complexity for extra (simpler) computation sequences**;
  flexibility of program algorithm to adapt to circumstances

Engaging with the ENIAC: Von Neumann versus Curry, a first summary

For von Neumann, the idea is indeed to use the simulation to test a model or theory, the **computer is mainly the bottleneck limiting the size of the sample**

- Sees high speed computing as way to introduce the **empirical back in mathematics**
- All metaphors (card = particle; computer = brain) serve to uphold the **analogy of a mathematician or physicist testing his theory**

For Curry, the simulation is but one of many **modifications of the program** possible, the interest lies in the relations and transformations between these, and **their impact on the computer's resources**

- Unit=program: "the problem of inverse interpolation is studied with reference to the programming on the ENIAC *as a problem in its own right* [our italics]."
- Instance of a general problem, *"Although from some points of view one way of formalisation is as good as any other, yet a certain interest attaches to the problem of simplification [...] In fact we are concerned with constructing systems of an extremely rudimentary character, which analyse processes ordinarily taken for granted."*

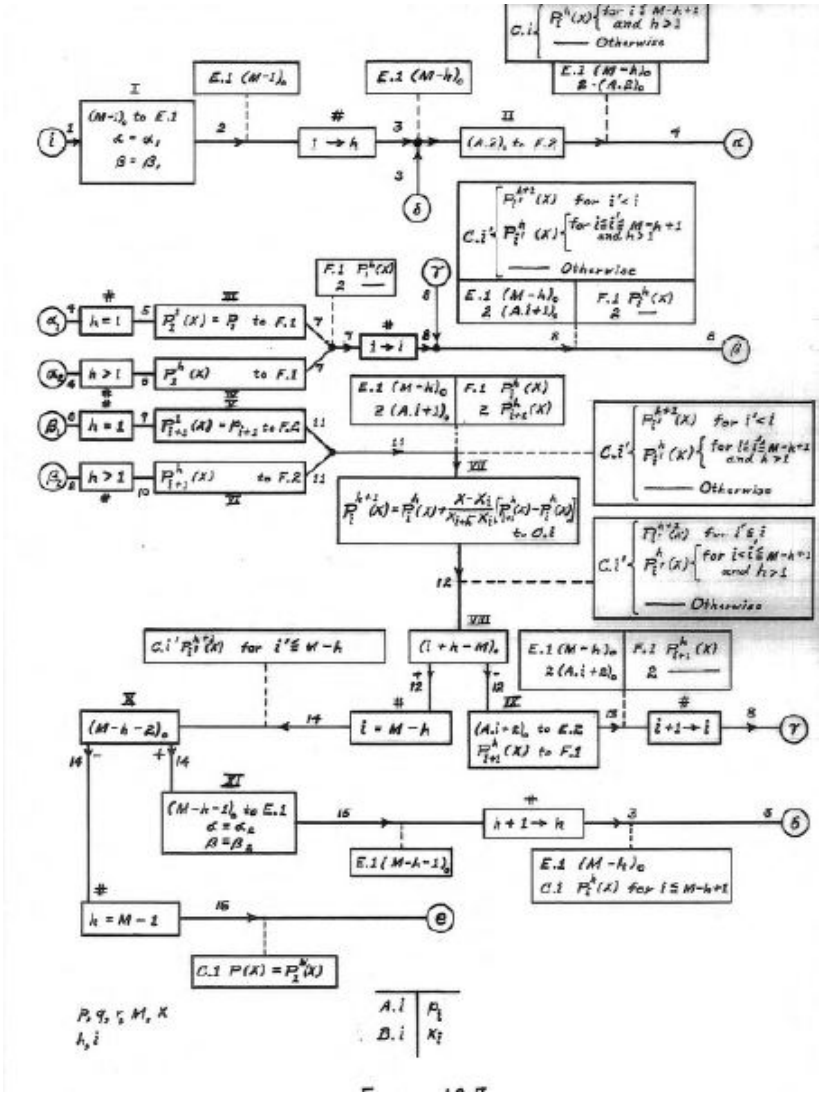
Von Neumann versus Curry: Impact on Programming and Machines

Rethinking of methods of **programming**

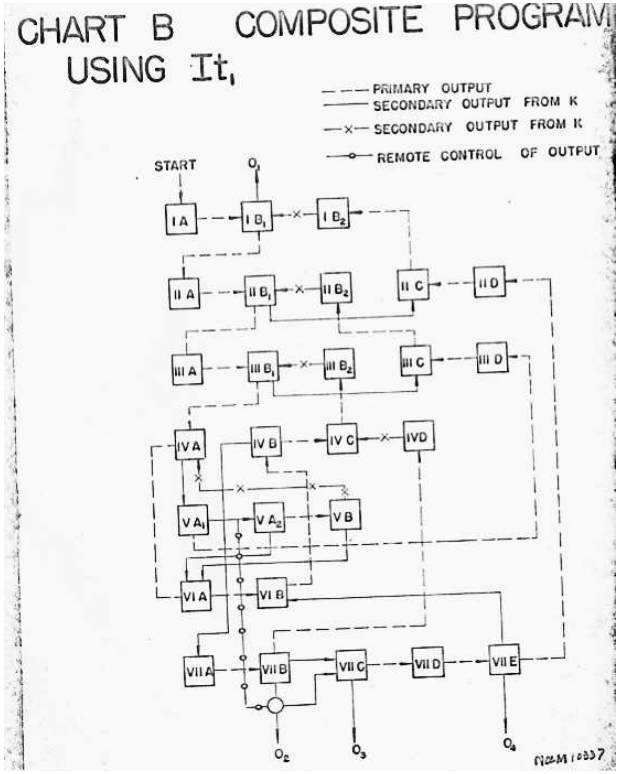
- Curry: method of **program composition**, finding how to (automatically) synthesize complex programs from simple ones
- von Neumann (and Goldstine): **flow charts** to analyze logical organization, *"The flow-diagram of a problem is prepared by the mathematician or physicist. [...] The next step in the preparation is the coding [...] This is routine."* (Manual Maniac)

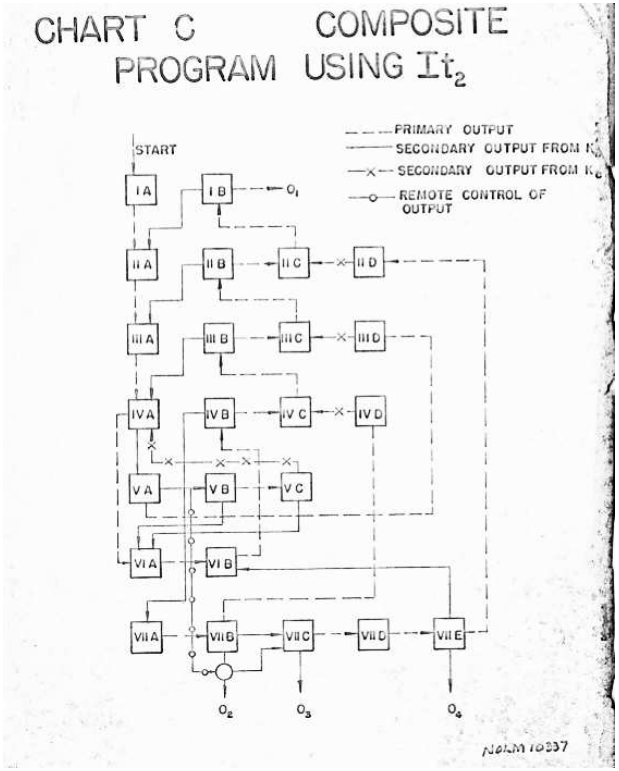
Rethinking **the machine**

- *"The Los Alamos problem, as originally programmed in 1945, exceeded the capacity of ENIAC. Many tricks were developed to expand the capacity without building extra equipment."*
  ENIAC rewired into a serial stored-program machine in 1948: *"ENIAC had sufficient flexibility to permit its controls to be reorganized into a more convenient (albeit static) stored-program mode of operation."*
- von Neumann, *Draft of the EDVAC* (1947) 'freezes' computer design into the so-called **'von Neumann architecture'**, a serial fetch-execute machine
- Curry, analyzes EDVAC-type machines (1949-50), and proposes a **RISC-like design** with provision for programming memory



Goldstine and von Neumann's flow diagrams, here for Lagrangian interpolation, mainly mapping the logical operation.





Curry's scheme for inverse interpolation, once with initial iteration control by primary program; and once with final iteration control by second program

Discussion

The 1000-fold speed-up the ENIAC provided to mathematicians spurned the development of:

1. New computational techniques:
   - Splines: Refinements and more complex controlling of known schemes because of the suitability of iteration schemes for ENIAC
   - Development of mixed techniques such as Monte Carlo, between the deterministic and stochastic, for the study of systems that are neither small nor continuous, and that now become implementable
   - ...
2. Ways of programming
3. New computer architectures

Does the numerical method add a representational layer? Yes, at least in a way. In the development of interfaces between man and computer, numerical methods are not passive programs, but active actors, shaping ideas on programming and even on hardware.