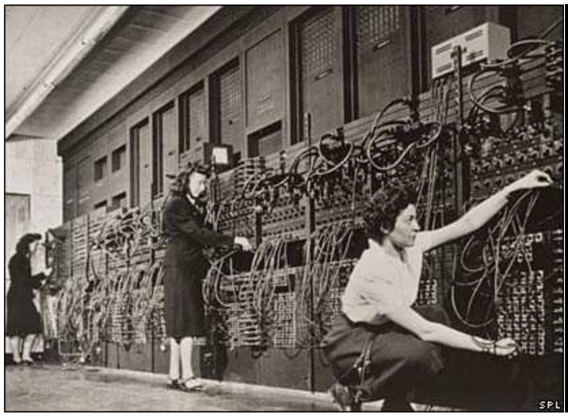
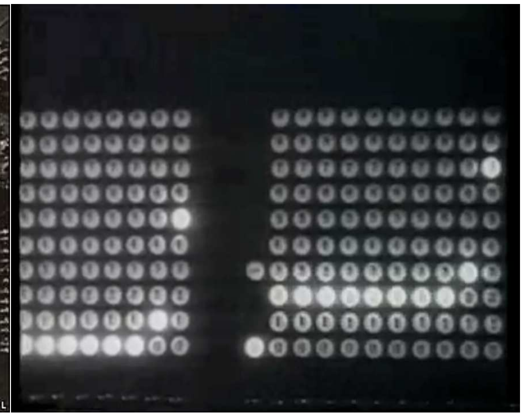
Exploring the universe of mathematics. Computation, experimentation and exploration in computer-assisted math





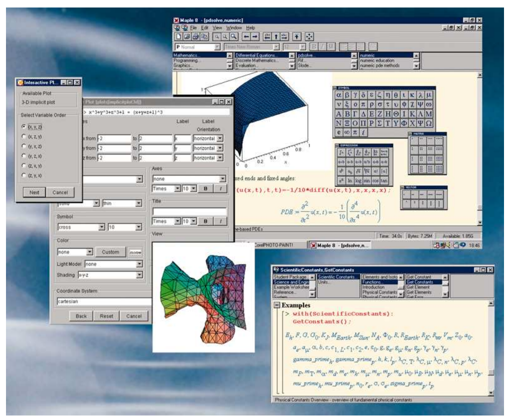
```
(define (initial-vec seq l n)
  (if (= l 0)
      seq
      (initial-vec (vector-append seq (vector (random n))) (- l 1) n)))
(initial-vec #(0) 300 2)


(define (add-period diff-periods n-period)
  (define (in-list? item seq)
    (cond ((null? seq) #f)
          ((equal? item (car seq)) #t)
          (else (in-list? item (cdr seq)))
          )
    )
  (cond ((= n-period (- 1)) diff-periods)
        ((in-list? n-period diff-periods) diff-periods)
        (else (append (list n-period) diff-periods))))

(add-period '(6) 1)

(define (determine-max r-max n-max?)
  (cond ((>= r-max n-max?) r-max)
        (else n-max?)))


(define (tag-it-vec count words v word ref-word ref-count)
  (define (deletion v word)
    (vector-tail word v))
  (define (produce words v word)
    (let ((appendent (vector-ref words (vector-first word))))
          (if (< (+ (- (vector-length word) v) (vector-length appendent)) 1) #()
              (deletion v (vector-append word appendent)))))
  (define (halt? word)
    (equal? word #()))
  (define (period? w1 w2)
    (equal? w1 w2))
  (define (period count ref-count)
    (- count ref-count))
  (let ((production (produce words v word)))
    (case (modulo count 1000000)
      ((0) (display count) (newline)(display (vector-length word)) (newline)))
    (cond ((period? word ref-word) (list count 0 1 (period count ref-count)))
          ((halt? production) (list count 1 0 (- 1)))
          ;((= count 100000) 1)
          ((= 0 (modulo count 1000)) (tag-it-vec (+ count 1) words v production word count))
          (else (tag-it-vec (+ count 1) words v production ref-word ref-count))
          )
    )
  )
```



Liesbeth De Mol
Centre for Logic and Philosophy of Science, Belgium
elizabeth.demol@ugent.be

First, some publicity.....

Turing in Context II: Historical and Contemporary research in Logic, computing machinery and AI

10-12 October 2012

http://www.computing-conference.ugent.be/tic2

Brussels

Keynotes: S. Barry Cooper, Leo Corry, Daniel Dennett, Marie Hicks, Maurice Margenstern, Elvira Mayordomo, Alexandra Shlapentokh, Rineke Verbrugge

Intro.

Introduction

⇒ **Motivation:** The increasing use of the computer in math seems to go hand-in-hand with growing significance idea "experimentation" and "exploration" in math – "computers [are] changing the way we do mathematics" (Borwein, 2008)

⇒ **Extent impact??**

- Mathematics proper
- Philosophy of Mathematics

⇒ ... and their interactions

Approach(es)

(General) approach

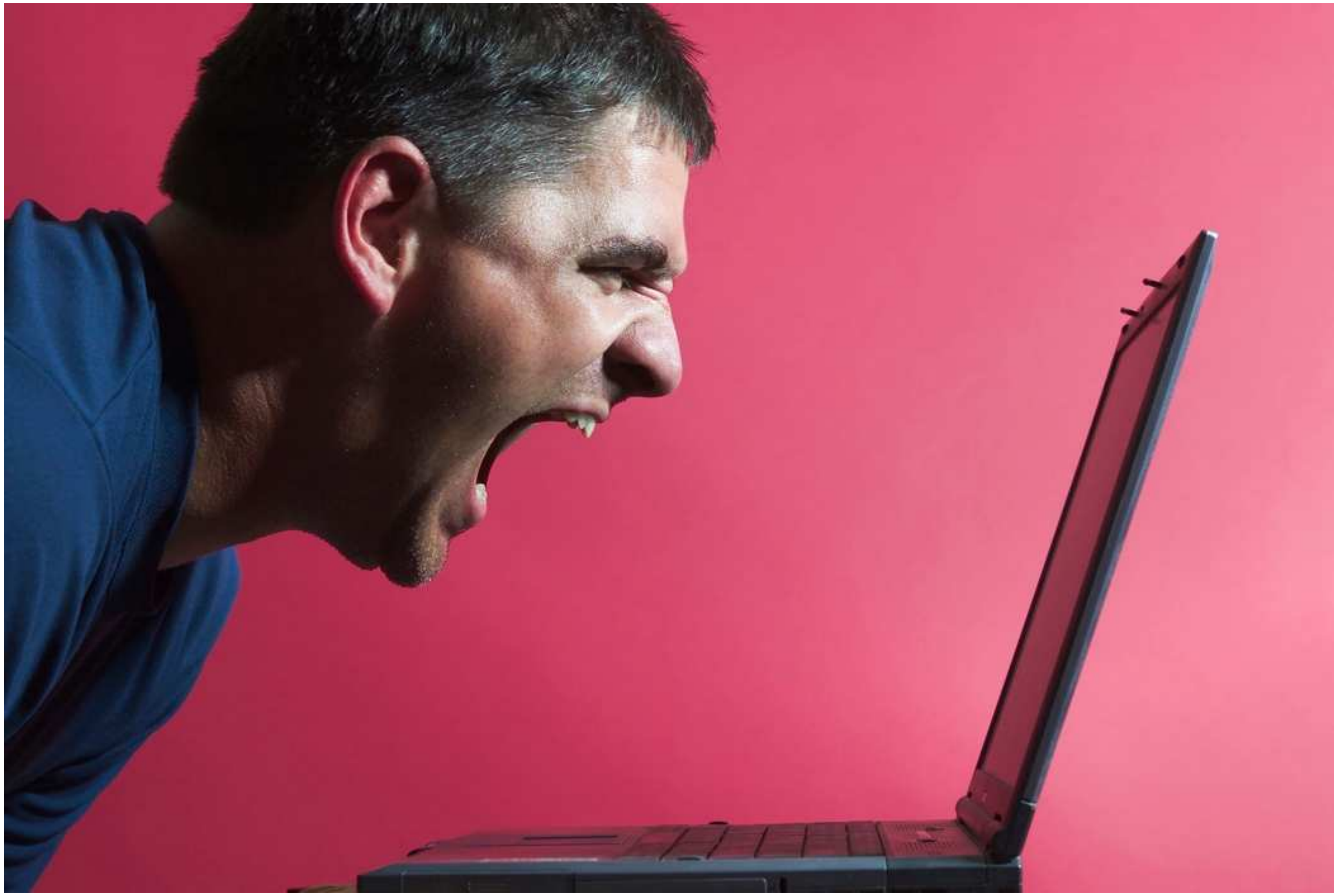
$\Rightarrow$ Bottom-up – and see where one gets

- Take computer *seriously* – as a **medium** (Kittler, 1985):

  ”*Media are no tools. Far more than things at our disposal they constitute the interaction of thinking and perception – mainly unconsciously.* (Carlé, 2010)

- Study mathematical **practice(s)** that is *really* guided by that practice $\rightarrow$ Study “gory” details of (history of) computer-assisted math + no π-in-the-sky-phil-of-math (also phil of math has a history!)

Taking the computer seriously....



Taking the computer seriously – two classical "myths"

- "*Another argument that continually arises is that* **machines can do nothing we cannot do ourselves**, *though it is admitted that they can do many things faster and more accurately. The statement is true, but also false. It is like the statement that, regarded solely as a form of transportation, modern automobiles and aeroplanes are no different than walking.* **[T]hus the change by six orders of magitude in computing have produced many fundamentally new effects that are being simply ignored when the statement is made that computers can only do what we could do ourselves if we wished to take the time**" (Hamming, 1965)

- "'**computers can only do what they are told to do**'. *True, but that is like saying that, insofar as mathematics is deductive, once the postulates are given all the rest is trivial. [...]The truth is that in moderately complex situations, such as the postulates of geometry or a complicated program for a computer,* **it is not possible on a practical level to foresee all of the consequences**" (Hamming, 1965)

Taking the computer seriously....

Study of 'experimental' computer-assisted math

⇒ Taking into account "material" and "social" changes of computer (changes in architecture, programming techniques, etc) in a study of computer-assisted math to detect global changes

⇒ Attention for four (intrinsically related) core features of CaM:

- Time-squeezing
- Space-squeezing
- Internalization (programmability)
- Mathematician-computer interactions (distribution of information and its processing during and after experimentation)

The question is not 'what is experimental math' in the context of CaM but rather 'What changes in (experimental) math' in the last 60 years?

⇒ How does the 'experimental' set-up change?

⇒ How does the M-C interaction change?

⇒ How are the mathematician's views on (experimental) math affected?

⇒ etc

Experimental math?

The Lehmers view on experimentation and CaP (in a nutshell)....

- ⇒ "[The first school of thought is concerned with] the improvement of highways between the well-established parts of mathematics and the outposts of the realm [favoring] the extension of existing methods of proof to more general situations" [The second school is concerned with] "the establishment of new outposts [...] This school favors *explorations* as a means of discovery" (Lehmer,1966)
  ⇒ Exploration makes possible math as an experimental science (but experimentation *does not* reduce to exploration: generation + exploration)

- ⇒ "[T]he most important influence of the machines on mathematics should lie in the opportunities that exist for applying the experimental method to mathematics."

- ⇒ Exploration and experimentation *not* specific for CaM!!

Experimental math – Four apps in time

I The Lehmer-ENIAC experience (+/- 1947)

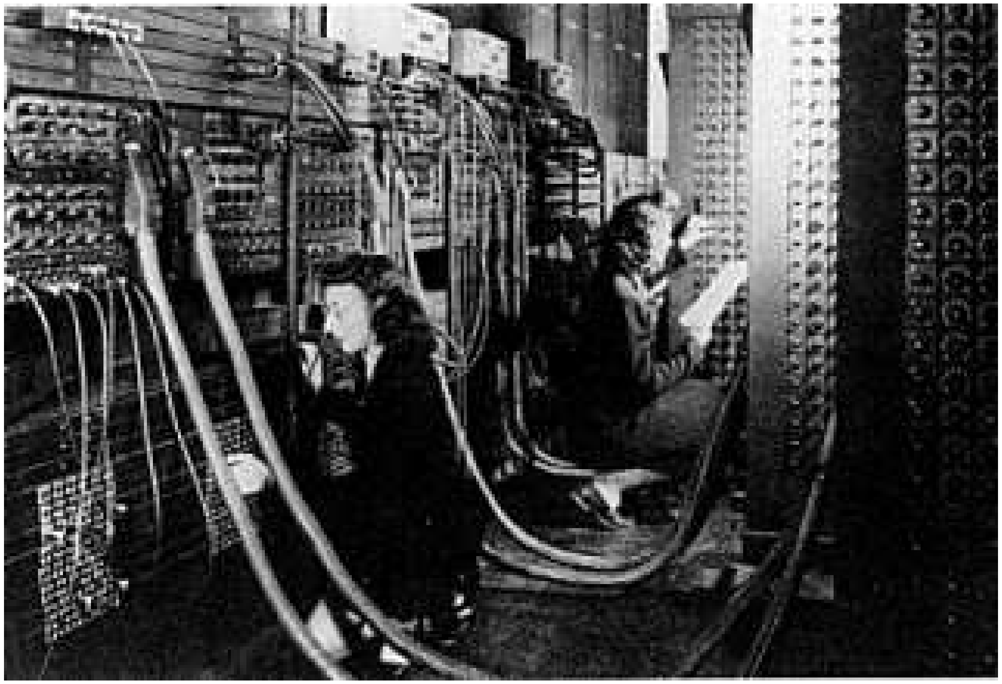
II Mandelbrot and his set (+/- 1980)

III The case of the Busy Beaver (+/- 1980, 1985)

IV Wolfram's new kind of science (1985; 2002)

Case I: The Lehmer-ENIAC experience

The ‘behemoth’ ENIAC



- ENIAC, The Electronic(!) Numerical Integrator And Computer
- Local and direct programming method: “The ENIAC was a son-of-a-bitch to program” (Jean Bartik)

  “The original “direct programming” recabling method 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” (Fritz, 1994)

- BUT, programmable + extremely fast for that time

The Lehmers and the first extensive number-theoretical computation on the ENIAC (joint research with M. Bullynck)

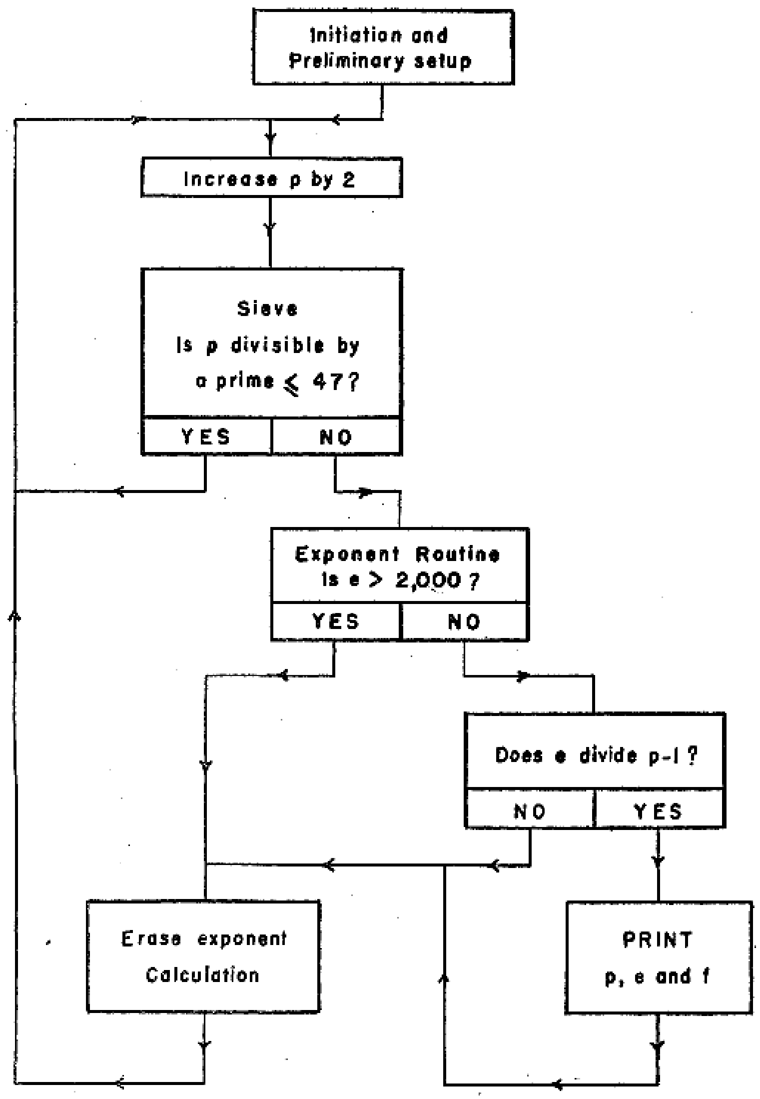
- "*I think what's particularly interesting about the number theory problem they ran was that this was a difficult enough problem that it attracted the attention of some mathematicians who could say,* **yes, an electronic computer could actually do an interesting problem in number theory**" (Alt, 2006)
- Exceptions to a special case of the converse of Fermat's little theorem If n divides $2^n - 2$ then n is a prime
- **Goal I** Testing the machine
- **Goal II** Finding composite numbers to generate tables of primes
- **Goal III** Finding mistakes in Kraitchik's table of exponents (up to $p \leq 300,000$)
- **Goal IV** Exploration of prime number tables in number theory

How was ENIAC used to compute composite numbers?

- The ENIAC was used to determine a list of exponents e of 2 mod p, i.e., the least value of n such that $2^n \equiv 1 \bmod p$ with p prime
- These exponents can be used to determine composite numbers of the form $2^{pq} - 2$ through the theorem:

  Theorem 1 *If p and q are odd distinct primes, then $2^{pq} - 2$ is divisible by pq if and only if p - 1 is divisible by the exponent to which 2 belongs modulo q and q - 1 is divisible by the exponent to which 2 belongs modulo p*
- Compute small numbers to compute big numbers
- A sieve was implemented on the ENIAC to determine primes relative to the first 15 primes, thus making use of the ENIAC's parallelism. The last prime p processed, after 111 hours of computing time, was $p = 4,538,791$ (Kraîchik hand-made table only to 300,000!)
- Eratosthenes's Sieve:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \ldots \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \ldots \end{pmatrix}$$


Initiation and Preliminary setup
Increase p by 2
Sieve
Is p divisible by a prime ≤ 47?
YES
NO
Exponent Routine
Is e > 2,000 ?
YES
NO
Does e divide p-1 ?
NO
YES
Erase exponent Calculation
PRINT
p, e and f

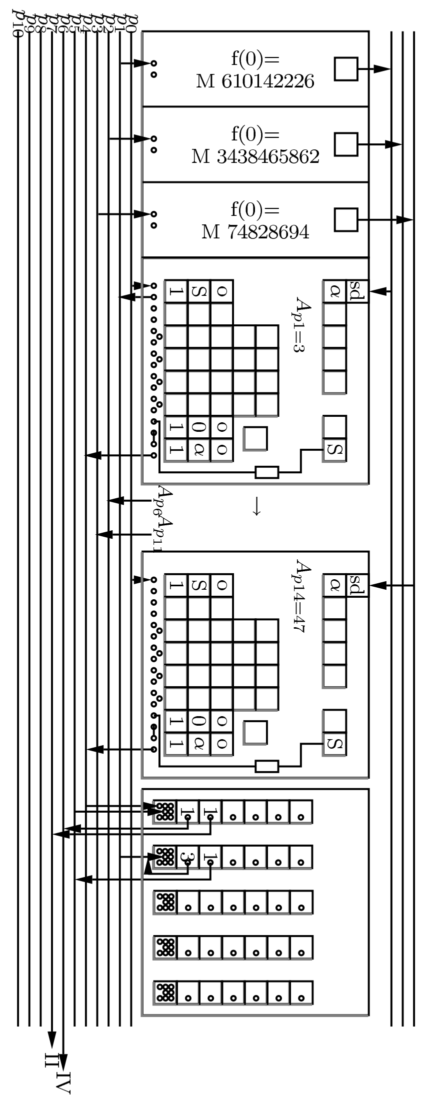

f(0)=
M 610142226
f(0)=
M 3438465862
f(0)=
M 74828694
A_{p1}=3
A_{p14}=47
II
IV

TABLE OF COMPOSITE SOLUTIONS n OF FERMAT'S CONGRUENCE $2^n \equiv 2 \pmod{n}$ AND THEIR SMALLEST PRIME FACTOR p

n	p	n	p	n	p
100463443	7577	312773	3541	558011	6449
618933	4729	413333	6067	940853	503
860997	9649	495083	1987	120296677	229
907047	5023	717861	1013	517021	2341
943201	5801	111202297	5273	838609	433
101152133	5807	370141	883	121062001	1201
158093	3673	654401	6101	128361	6961
218921	8713	112032001	4001	374241	6361
270251	9001	402981	3061	121472359	4409
276579	6163	828801	6133	122166307	739
954077	1597	844131	3067	396737	2857
102004421	2381	113359321	761	941981	337 · 491
443749	4049	589601	331 · 571	123330371	691
678031	3583	605201	7537	481777	3881
690677	2069	730481	433	559837	4177
690901	5851	892589	919	671671	9631
103022551	6121	114305441	6173	886003	1187
301633	7873	329881	7561	987793	709
104078857	6679	469073	3089	124071977	2089
233141	2441	701341	1229	145473	397
524421	5903	842677	2459	793521	4561
105007549	1033	115085701	1801	818601	2281
305443	2833	174681	773	125284141	4231
919633	4603	804501	5381	686241	6473
941851	1051	873801	1051	848577	2897
106485121	7297	116090081	6221	126132553	5023
622353	433	151661	7621	886447	6793
743073	1699	321617	5393	127050067	5347
107360641	2161	617289	2357	710563	9787
543333	4889	696161	2161	128027831	11161
108596953	7369	998669	1459	079409	5437
870961	2609	117246949	1597	124151	2311
109052113	4993	445987	5419	468957	2927
231229	2699	959221	2053	536561	8017
316593	3697	987841	7681	665319	2383
437751	5231	118466401	1249	987429	4637
541461	6043	119118121	2729	129205781	6563
879837	2707	204809	2383	256273	739
110135821	3967	261113	4657	461617	10177
139499	6427	378351	911	524669	2939

Lehmer's way of dealing with the machine...

- **Computing from the machine's point of view** "The method used by the ENIAC to find the exponent of 2 modulo p differs greatly from the one used by human computer" (Lehmer, 1949)
  "In contrast, the ENIAC was instructed to take an "idiot" approach"

$$\Gamma_1 = 2, \Gamma_{n+1} = \begin{cases} \Gamma_n + \Gamma_n & \text{if } \Gamma_n + \Gamma_n < p \\ \Gamma_n + \Gamma_n - p & \text{otherwise} \end{cases}$$

  Only in the second case can Γ_{n+1} be equal to 1. Hence this delicate exponential question in finding $e(p)$ can be handled with 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)

- **Partial Internalization and heuristic program** "The "next value of p" [i.e. the next prime] presents an interesting problem to the ENIAC. [Circumstances] prevented the introduction [of] punched cards. [...] This means that the ENIAC should somehow compute its own values of p. To this effect a "sieve" was set up which screened out all numbers having a prime factor ≤ 47. [Else there is a need for] "much outside information [introduced] via punched cards [...] to be prepared by hand in advance" + 25 out of 11336 eliminated by hand

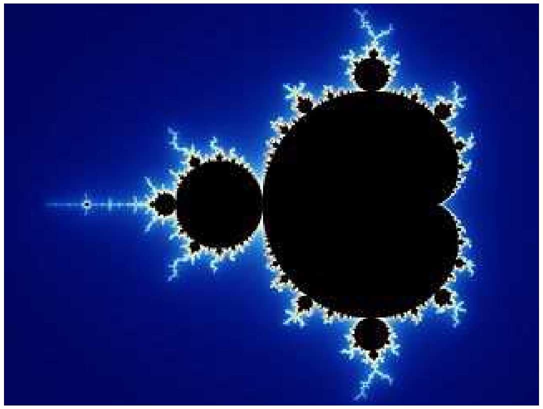
Lehmer's way of dealing with the machine...

- "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"
- **External/human processing** "The list of exponents furnished by ENIAC is sufficient for the extension of the table of composite numbers n dividing $2^n - 2$ to 10^9 and beyond. However the list presented herewith extends only from 10^8 to 2×10^8. **The labor of producing these composite numbers is still considerable.**"
- **One long computation without 'responsive interactions'**: "*[Lehmer] had programmed the problem and run it on ENIAC, with J. Mauchly serving as "computer operator", during the three-day weekend of July 4, 1946.* **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**." (Alt, 1972)

Lehmer's ENIAC 'Experiments'? Problem: gap between speed processing vs. retrieval and storage; restrictions on programmability (no intermediary language), processing power and memory (external storage and retrieval) and availability

⇒ **Largely discontinuous process of "experimentation"** Separated phases of the experiment distributed over human and machine, with human doing most of the 'exploration'

Case II: Mandelbrot and his set



DEC-VAX-11/780 ("star")

- Follow up of the PDP-11. Release dat: October 25, 1977; first one installed at CERN
- **Speed** 3.4 MhZ (10^6 herz); 500,000,000 instructions/second
- **Memory** From 128kb up to 8 MB static memory! 1MB static and 4k ram
- **i/o devices** Tektronix terminal, versatec printer
- **Programming** VMS operating system ("starlet"), with GUI and graphics support (!); support for multiple programming languages (FORTRAN, COBOL, BASIC? PASCAL, etc)

Case II: Mandelbrot and his set

- Late 70s interested in the theory of rational maps of the complex plane (knowledge of work by Fatou and Julia) $\rightarrow$ "playing around" with quadratic Julia sets defined through iteration $z \rightarrow z^2 + c, c, z \in \mathbb{C}$
-
$$E_c = \{z_0 : |z_n| \rightarrow \infty\},\ z_n = z_{n-1}^2 + c$$
$$P_c = \{z_0 : z_0 \notin E_c\}$$
  Julia set for c is the boundary of E_c.
  Fundamental dichotomy for Julia sets: connected and disconnected sets.
- For certain c, some points z always converge to a finite stable cycle of size $n \rightarrow$ attempt to classify Julia sets according to n (only connected sets!)
- **Fact:** The prisoner set P_c for $z \rightarrow z^2 + c$ is connected iff the orbit $0 \rightarrow c \rightarrow c^2 \rightarrow c^2 + c \rightarrow \ldots$ remains bounded.
- 1980: Exploration of the map $c \rightarrow c^2 + c$
- The Mandelbrot set:
$$M = \{c \in \mathbb{C} | c \rightarrow c^2 \rightarrow c^2 + c \rightarrow \ldots \text{remains bounded}\}$$

Case II: Mandelbrot and his set

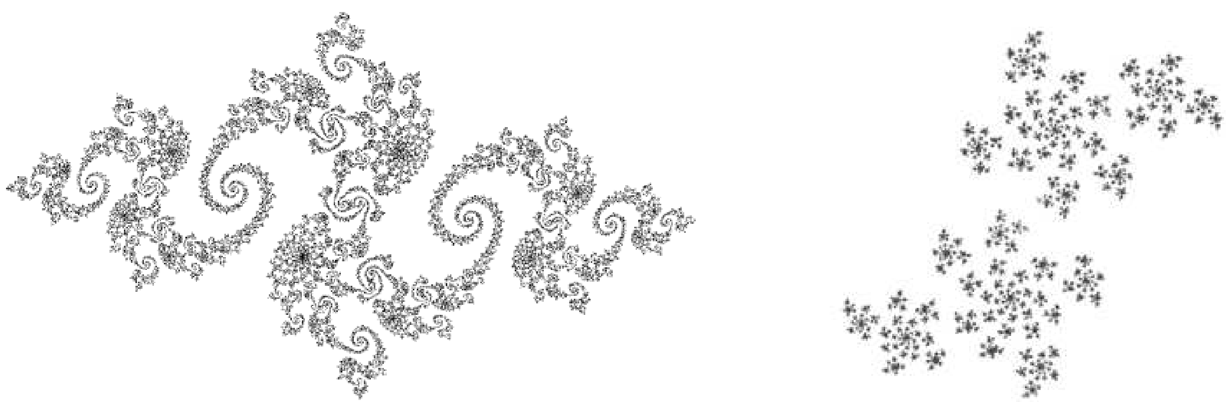


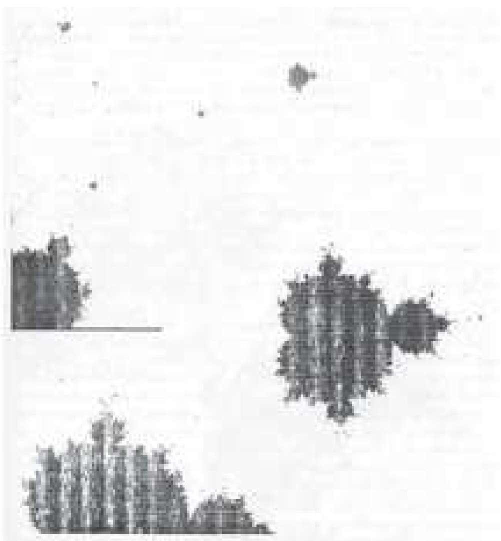
Figure 1: A connected and a disconnected Julia set

From seeing and knowing to discovering and results (1)

Visualizations as a means to find new conjectures, ideas, etc

- M as a road map for Julia sets: Classification of Julia sets with stable finite cycles n as smaller and smaller "sprouts" of M
- Connection between left-side of the cardioid and period-doubling bifurcations
- Apparent "specks of dirt" on print-out are "real" $\rightarrow$ zoom-in reveal "island whose shape is like that of M, except for a non-linear deformation. Each island is, in turn, accompanied by sub-islands, doubtless ad infinitum" (Mandelbrot, 1980)
- Julia sets with c's in specks of dirt + connectedness of these J-sets + theory of bifurcation: Conjecture M as a connected set (proven in 1982 – Douady & Hubbard)

From seeing and knowing to discovering and results (2)



Possibility of machine error “early originals looked awful: filled with apparent specks of dust that the Versatic printer produced [T]his complicated matters. But for the skilled, meticulous, and tireless observer that I was, mess was not a reason to complain but a reason to be particularly attentive”

Increased internalization and heuristic methods

Use of encirclement method? Computation on millions of internally stored data. You are never sure about the outcome (undec. M-set)

“By a theorem Julia and Fatou, those Julia sets are connected. Therefore the broken-up appearances is necessarily due to the discrete variables used in computation. These graphs were important to my thinking because they sufficed to show hat the broken-up earlier early M set pictures were compatible with connectedness”

From seeing and knowing to discovering and results (3)

Experimentation as Machine-aided, human-directed visual exploration

- ⇒ "When seeking new insights, I look, look, look and play with many pictures (One picture is *never enough*)"
- ⇒ Made possible through advanced machine technology: Internal machine "processing/translation" of "low-level" data to humanly practical format, i.e., graphical picture of M to explore
- ⇒ **pictures as an interface** Terminology and concepts directly inspired by pictures → result of interfacing between human and machine "processing"
- ⇒ "The 'fate' that drove me to revive the theory of iteration, first chose me to **reinvent the role of the eye** in a field, mathematics, where it and explicit computation had become anathema, about as unwelcome as they could possibly be"
- ⇒ My goal [with my 1980 paper on M] was to revive experimental mathematics by reporting observations triggering new mathematics"
- ⇒ ≈ Lehmer: "Yesterday, "generality at all cost" was in the saddle. Today, "special" problems are more readily recognized as compelling."

Mandelbrot's Experiments? Increased speed and memory; stored programming + programming language; (clumsy) printing devices

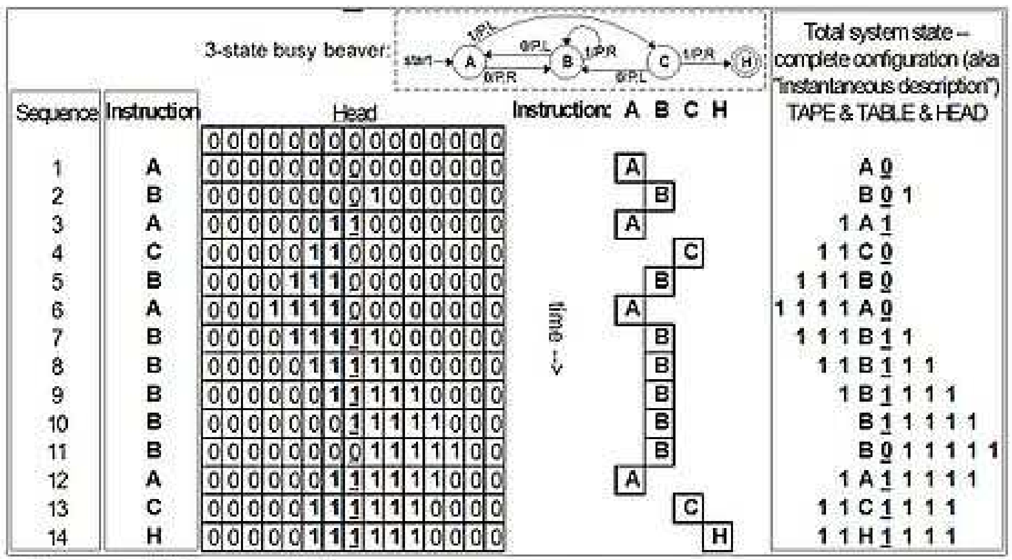
⇒ **"Space squeezing"**: "low-level" data are no longer humanly practical – internal machine computations "represented" in a humanly digestible way $\rightarrow$ uncertainty results

⇒ **"Time squeezing"**: Many smaller "experiments" and the flow of informations during this process of "experimentation" squeezed in a "reasonable" time-frame $\rightarrow$ increased involvement with the machine $\Rightarrow$ **Exploration vs generation? (e.g. zooming-in)** - ($\approx$ "real-time" manipulations of the M-set)

⇒ **More "continuous" process of human-machine "experimentation"**

⇒ **Increased interaction** during the "experiment": mixing of computation, exploration and interpretation in a process

Case III: Brady and Busy Beavers



Progress of the computation (state-trajectory) of a 3-state busy beaver

Case III: Brady and Busy Beavers.

Machine used? "A Turing machine simulator written in **a machine independent form of BASIC** is available from the author upon request"

The Busy Beaver problem Determine for any class of Turing machines $\mathrm{TM}(m, 2)$ with m states and 2 symbols the maximum number of 1s $\Sigma(m)$ respectively the maximum number of computation steps $S(m)$ left on the tape by some $T \in \mathrm{TM}(m, 2)$ that halts when started from a blank tape. First formulated and proven recursively unsolvable by Rádo, 1962.
$\Rightarrow$ Early on, computer-assisted studies and proofs (!) of the Busy Beaver problem for particular m:
Some results

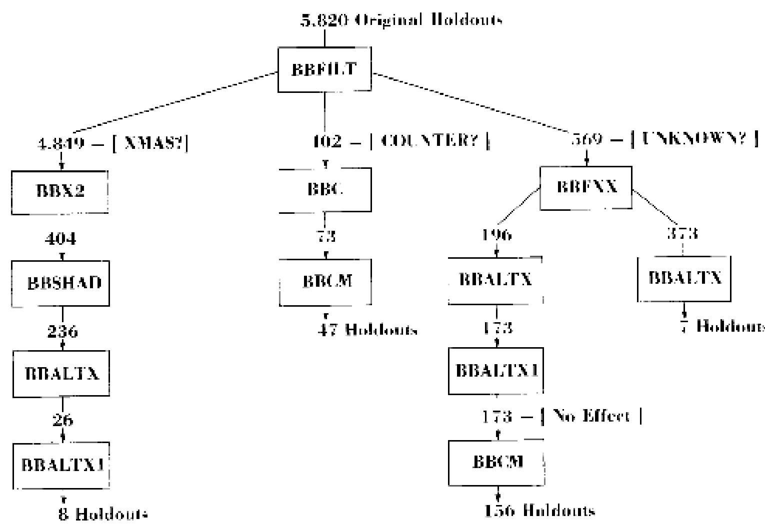
- $S(2) = 4, \Sigma(2) = 2$, Rádo (1962)
- $S(3) = 21, \Sigma(3) = 6$, Rádo and Lin (1965)
- $S(4) = 107, \Sigma(4) = 13$, **Brady (1983)** and Kopp (cited by Machlin and Kopp/Stout (1990))
- $S(5) =?47,176,870, \Sigma(5) =?4098$, Marxen and Buntrock (1990)
- $S(6) = 2.5 \text{ x } 10^{2879} \Sigma(6) > 4.6 \text{ x } 10^{1439}$, Terry and Shawn Ligocki (2007)

Computing Busy Beavers (1) (Brady, 1966)

- Notation instruction: (state, number read, number printed, move left/right, next state)
- The number n of Turing machines $T \in T(m, 2)$: $n = (4m+1)^{2m}, m = 4, n = 6,975,757,441$
- Approach: Determine the set of halting machines by reducing the number of "hold-outs" to 0.
- Brady's 1966 reduction to 5820 hold-out: Tree normalization and backtracking
  1. Eliminate machines for which (1, 0, 1/0, L/R, halt); idem for (1, 0, 1/0, R/L, 1)
  2. Exclude the symmetrical left-right machines and retain the right-left machines (or vice versa).
  3. Generalization idea 1 (backtracking): prove that machine is in infinite loop by showing with backtracking that halting state cannot be reached $\rightarrow$ Generation of instructions as they are needed (e.g.: (1, 0, 1, R,2), only 8 out of the 16 possible next instructions need to be generated .

Computing Busy Beavers (2) (Brady, 1983)

Internalization and exploration – **Identification of loops** (through exploration of hundreds of printouts) and **automated detection** of several types of infinite loops – human-machine collaboration



⇒ Parts of the 'proof' and its 'discovery' are done by machine – proof in-between computer and Brady (the typical unsurveyability problem)

⇒ "*[I]t must be remembered that the filtering [BBFILT] was a* **heuristic technique based upon experimental observation.**" → tentative classification based on the rate at which new squares are visited; "The proof techniques, embodied in programs, are entirely heuristic"

⇒ **Unpredictability + finite time** and the need of making decisions in finite time: "*As with all the heuristics we discuss, one must make some decision as to how long to run this technique before abandoning it.*"

Computing Busy Beavers (3)

- **Not one but many programs internalized** "More than 18 other programs were written, for various housekeeping purposes, simulating and displaying machine behavior, exploring other reduction and filtering possibilities etc. In all, at least 53 files were created and maintained for the project. Keeping track of **what resembled a large scientific experiment** became a major task in itself." ⇔ Lehmer's flowchart

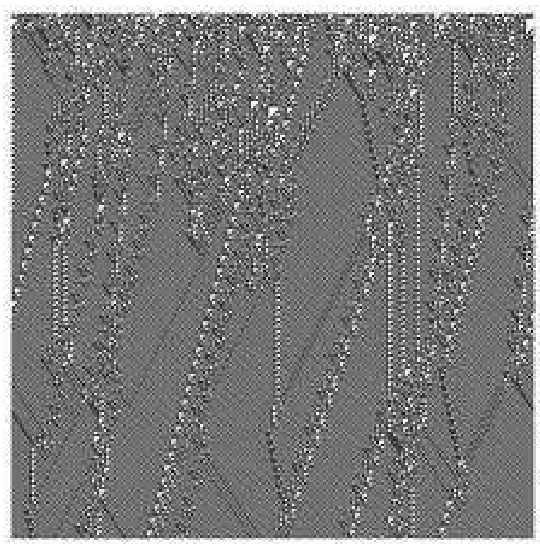
- **The problem of error** "While not all of the exploratory activities are reproducible, the runs [can] be reproduced, so that by utilizing the techniques described in this paper the proof can be corroborated. [...] Proofs of "correctness" of the programs used are not practical. Independent verification is the only means we currently have at our disposal."

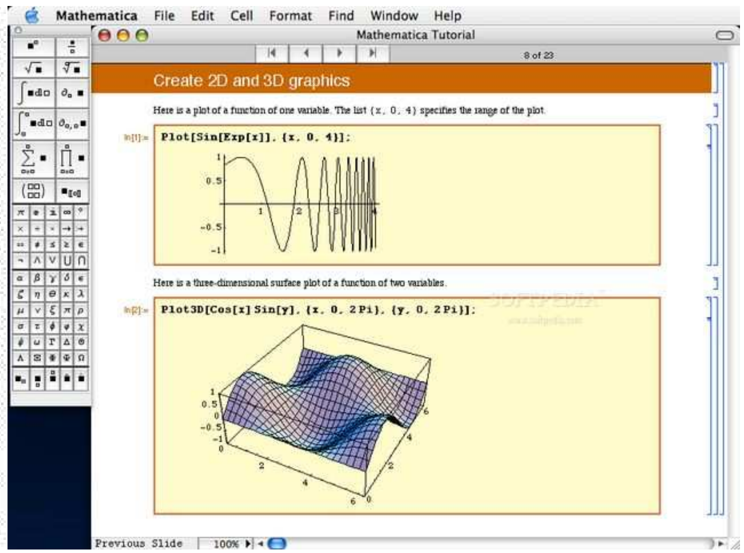
Brady's 'experiments'?

Increased speed and memory; portable programming language; (clumsy) printing devices; increased programmability

- ⇒ **Continuous and integrated process of human-machine "experimentation"**: Exploratory activities distributed between human and machine.
- ⇒ **Increased interaction and time-squeezing** Intertwinement of human and machine contribution. Proof within the interaction → Towards a human-computer collaboration

(Short) Case IV: Wolfram, Mathematica and a “new kind of science”





(Short) Case IV: Wolfram, Mathematica and a "new kind of science" (1984-1988)

- **Machine used?** the C language computer program;"CA: an interactive cellular automaton simulator for the Sun Workstation and VAX"; Connection Machine computer;...
- **Some technical (observational) results:** four classes of behavior; conjecture universality rule 110 ("*This paper covers a broad area, and includes many conjectures and tentative results. It is not intended as a rigorous mathematical treatment.*"); random number generator based on rule 30
- **Complex behavior simple programs:** "It is remarkable that such a simple system [rule 30] can give rise to such complexity. But it is in keeping with the observation that mathematical systems with few axioms, or computers with few intrinsic instructions, can lead to essentially arbitrary complexity. And it seems likely that the mathematical mechanisms at work are also responsible for much of the randomness and chaos seen in nature."
- **Complexity in physics** Undecidability and intractability in physics: "It is the thesis of this paper that [problems of computational irreducibility] are in fact common

⇒ **Before Mathematica:** Most of the basic results on CA already found

⇒ **Start development of a general theory inspired by computer science**

(Short) Case IV: Wolfram, Mathematica and a "new kind of science" (1988–now)

- **Mathematica (1988)**: "I first conceived of Mathematica because I needed it myself"
- "[T]he visionary concept of Mathematica was to create once and for all **a single system that could handle all the various aspects of technical computing**–and beyond–in a coherent and unified way." ⇒ Enter (again) generality!
- 2002: the long-awaited publication of "A new kind of science", based on theory of cellular automata as models for physical systems. Main method: "computer-based models and experiments"
- Connection Mathematica and "A new kind of science"?
- ≈ Maple and "Mathematics by Experiment" (Borwein and Bailey, 2004)

(Short) Case IV: Wolfram, Mathematica and a "new kind of science" (1988–now)

What is the significance of software like Mathematica and Maple for the development of "experimental mathematics"? (See e.g. Sorensen, 2010: fact-gathering vs. interactive exploration)

- "interactive exploration" is not the sole domain of Maple or Mathematica (See Cases)

⇒ Pre-programmed internalization and centralization of different aspects of "experimentation": statistical tools, graphics tools, special algorithms (user-friendliness)

⇒ **Time-squeezing** No wasting time on programming the tools; "real-time" manipulations and computations

⇒ **More continuous and integrated human-machine experiments**

⇒ **Increased (faster) interaction** Possibility of more "direct" interaction with the emulated/simulated objects studied.

⇒ Wider accessibility and integration of knowledge: development of "general" and "integrated" theories

Discussion

Discussion

- From the “behemoth” ENIAC to Mathematica/Maple: process of changing mathematician-machine interactions and ‘experiments induced by technical changes: from a discontinuous process of computer-assisted experimentation to a more continuous and integrated one
- (From a micro perspective) Change on the level of the method of experimentation: not one “smaller” experiment but many “phases” and “aspects” of experimentation integrated into one (“time squeezing” and “internalization”)

$\Rightarrow$ Affects the way an ‘experiment is set-up, the kind of experiment one can do, how much responsibility is for the machine, the kind of local problems one has to deal with, the methods one develops etc

Discussion: Many questions, thoughts...

- ⇒ Distributed computing? The internet? Social aspects of math and computing (mailing, blogs, publishing, etc)
- ⇒ **A(n) (computer) experiment in math?**? Mathematical (computer) experiment $\neq$ computation: Explicit integration of "pure" computations ("nature") with exploration, concept-formation, conjecturing, etc *and* heuristic and probabilistic programming ⇒ **Not reasoning *with* but *in* computer experiment**
- ⇒ **To what extend is the computer *really* changing math?** What is the difference between e.g. Brady's "explorations" and Gauss' "explorations"? Computers demands to re-think locally (e.g.. Mandelbrot's pecks of dirt; heuristic programs and uncertainty, etc)
- ⇒ "Progress" and the necessity of hiding the "source" (speeds-up)?
- ⇒ Recent Nature paper: "The vagaries of hardware, software and natural language will always ensure that exact reproducibility remains uncertain, but withholding code increases the chances that efforts to reproduce results will fail."
- ⇒ **A "paradox" of mathematician-computer interactions?** Growing distances between mathematician and physical computer and time-squeezing results in more direct and intertwined interactions that reflect upon our

thinking on “experimental math”

⇒ **The proof is in the process** “In any event, whenever [the] stage [of high baroque] is reached [in mathematics], the only remedy seems to me to be the rejuvenating return to the source: **the reinjection of more or less directly empirical ideas**.” (Von Neumann, 1947) ⇒ reinjection of time into mathematics as a fundamental question for computer-assisted math?