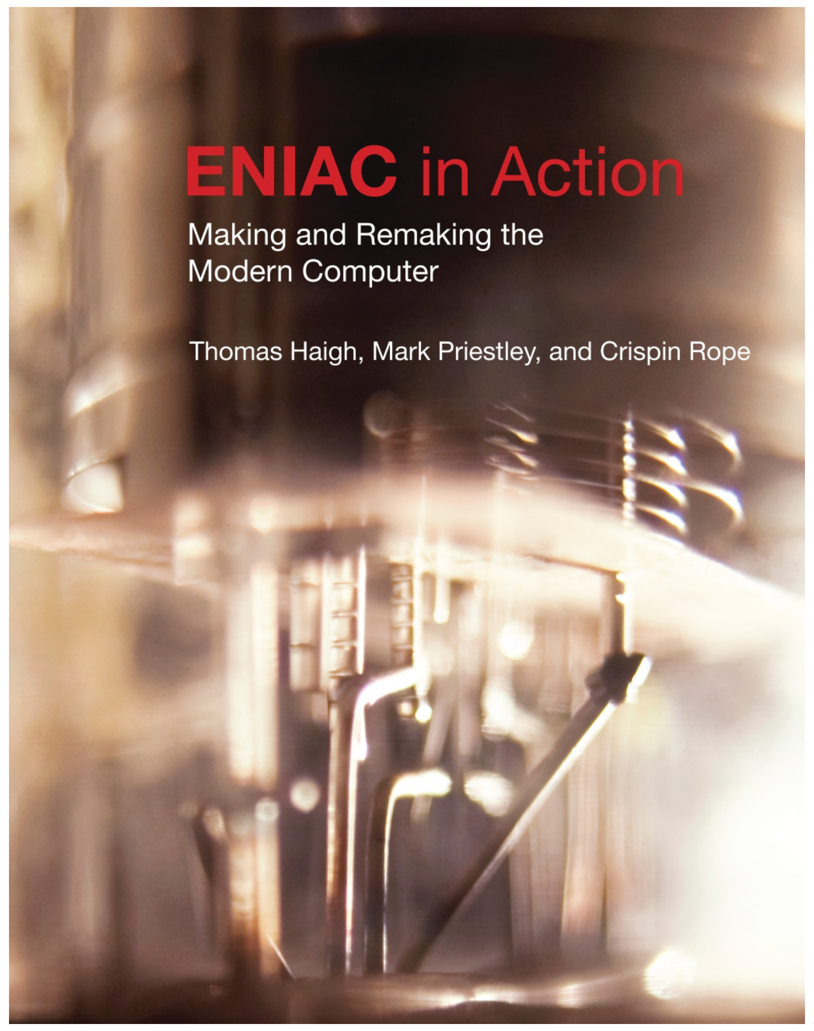
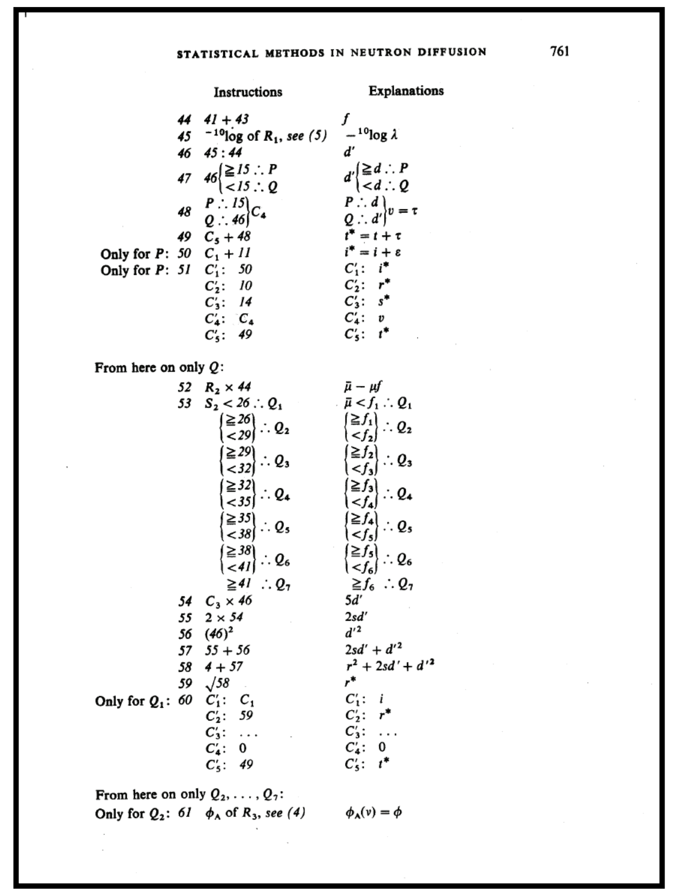
ENIAC Monte Carlo Flow Diagram, 1947

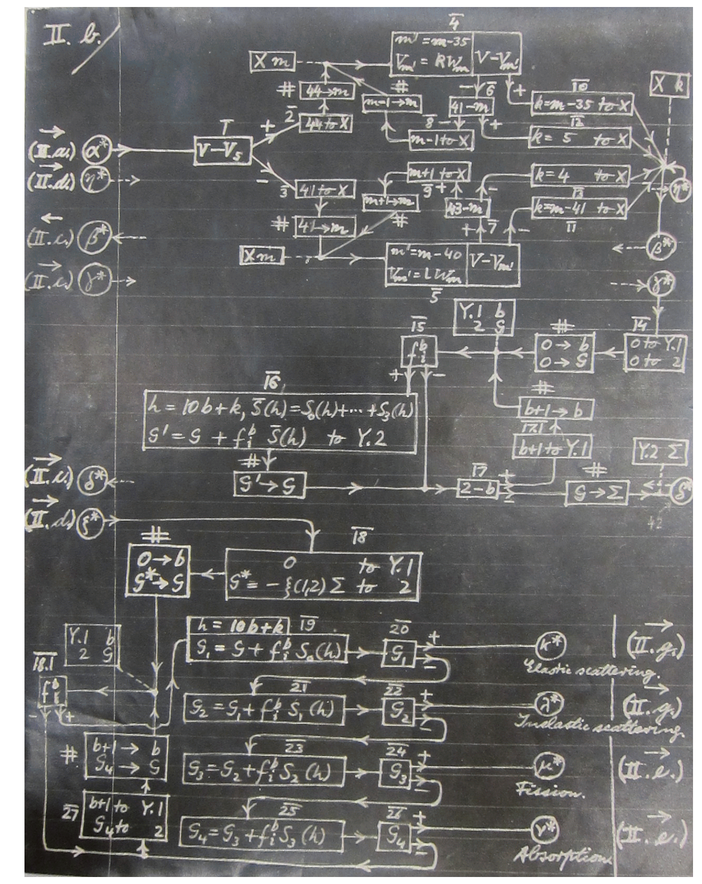


MIT Press, 2016

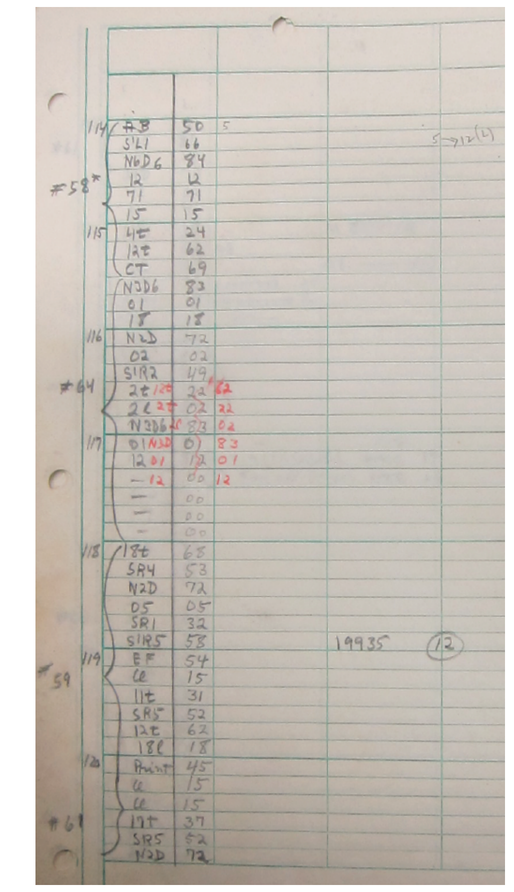
1



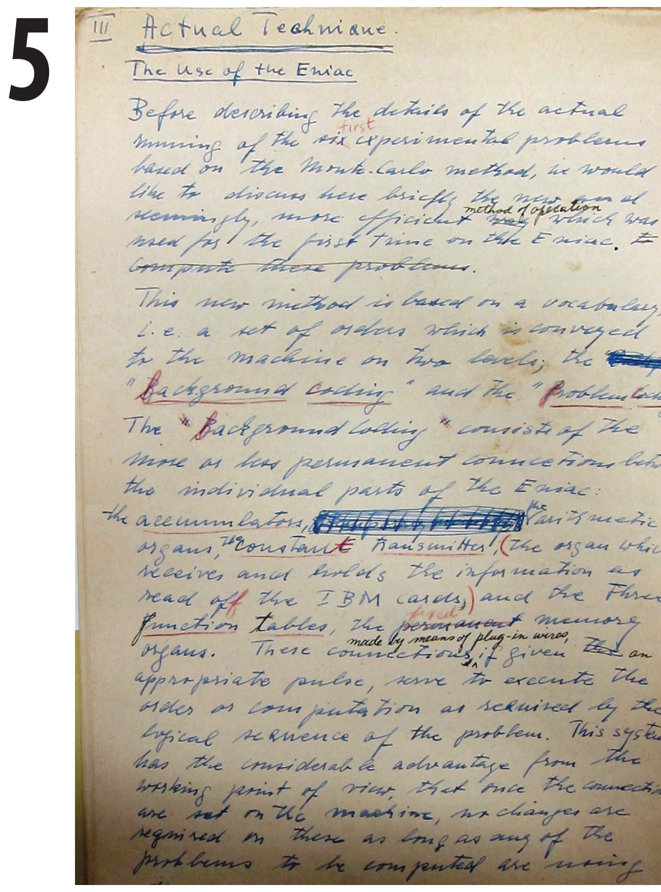
2



4



5



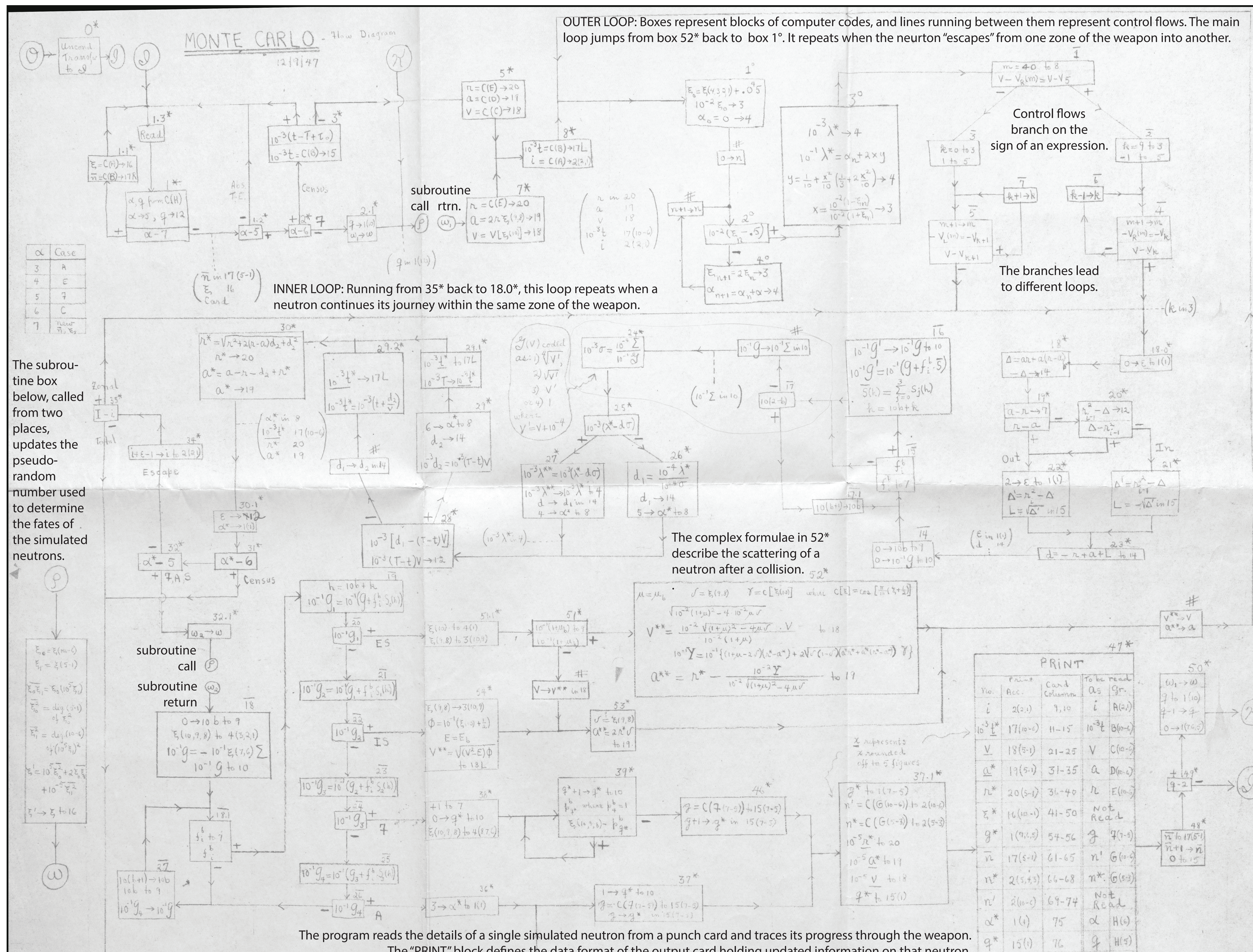
The main diagram (3) specifies the structure and operations of the Monte Carlo program run on ENIAC in March and April of 1948 to simulate the propagation of neutrons inside exploding atomic bombs. This was the first time modern code was run on any computer and the first electronic Monte Carlo simulation. John von Neumann came up with an original plan for the computation, in a letter sent to Los Alamos (detail as 1). He then drew up a draft flow diagram (detail as 2) which went through several revisions in collaboration with Klara von Neumann and Nick Metropolis before this version. We discovered a full listing (detail as 4) in Klara's handwriting for a revised version of the program run in late 1948. She described the procedures used in a report (first manuscript page as 5).

Boxes	Function
1* – 8*	Read a card and store neutron characteristics
1° – 4°	Calculate random parameter λ*
$\overline{1}$ – $\overline{7}$	Find neutron's velocity interval
18* – 23*	Calculate distance to zone boundary
$\overline{14}$ – $\overline{17.1}$, 24*	Calculate cross-section of material in zone
25* – 27*	Determine if terminal event is collision or escape
28* – 30*	Determine if a census comes first
31* – 35*	Discriminate between terminal events
Subroutine ρ/ω	Refresh random number
$\overline{18}$ – $\overline{27}$	Determine collision type
51* – 52*	Elastic scattering
53* – 54*	Inelastic scattering
36* – 39*, 46*	Absorption/fission
37.1*, 47* – 50*	Print card and restart main loop

Learn more at www.EniacInAction.com.

Images reproduced from the Library of Congress coutesy of Marina von Neumann Whitman.

3



MONTE CARLO Flow Diagram
12/9/47
OUTER LOOP: Boxes represent blocks of computer codes, and lines running between them represent control flows. The main loop jumps from box 52* back to box 1°. It repeats when the neurton "escapes" from one zone of the weapon into another.
Control flows branch on the sign of an expression.
subroutine call rtrn.
The branches lead to different loops.
INNER LOOP: Running from 35* back to 18.0*, this loop repeats when a neutron continues its journey within the same zone of the weapon.
The subroutine box below, called from two places, updates the pseudo-random number used to determine the fates of the simulated neutrons.
The complex formulae in 52* describe the scattering of a neutron after a collision.
subroutine call
subroutine return
PRINT
The program reads the details of a single simulated neutron from a punch card and traces its progress through the weapon.
The "PRINT" block defines the data format of the output card holding updated information on that neutron.