Monte Carlo Second Run Code: Reconstruction and Analysis

Mark Priestley & Thomas Haigh
www.EniacInAction.com
January 2016

This document provides a detailed and comprehensive reconstruction of the ENIAC code used by Nick Metropolis and Klara von Neumann during the "Second Run" Monte Carlo calculations from October 22 to November 7. It provides technical background for the work we present in the paper "Los Alamos Bets on ENIAC"[1] and in chapters 8 and 9 of the book *ENIAC In Action: Making and Remaking the Modern Computer* (MIT Press, 2016).

The evolution of the program between the First Run in April/May 1948 and the Second Run is discussed in the companion document, "Analysis of the Monte Carlo Computations, 1947-48," also available from www.EniacInAction.com. It is clear that the changes made between the spring and the autumn of 1948 improved the code in several ways, but they did not introduce any fundamentally new capabilities either in the physical model or in the style of the code used.

1 Sources

We have four principal sources for the details of the Second Run program. These are:

- An unnumbered, undated flow diagram.[2]
- An undated flow diagram, with numbered operation boxes.[3]
- The program code for the Second Run calculations.[4]
- The report *Actual Running of the Monte Carlo Problems on the ENIAC*.[5]

There is a very close relationship between the code and the numbered flow diagram, which will therefore be referred to in the remainder of this report as the "Second Run flow diagram". The unnumbered flow diagram appears to be a preliminary version of this, and will be referred to as the "draft flow diagram". *Actual Running* provides a textual description of the program that exhibits a high degree of consistency with the computation described in the Second Run code and flow diagram.

The computation described in the Second Run flow diagram can be thought of as the basic form of the program. The code listing includes some additions and variants, including a substantial piece of programming referred to as the "special process" applicable only to certain problems within the run. As mentioned in *Actual Running*, the special process is not described on the Second run flow diagram. However, closely related functionality is described in flow diagrams for the First Run[6] and a later flow diagram[7] and this has enabled this section of code to be analyzed and annotated as well.

We have used this material to construct a single annotated version of the code of the Second Run program. The archival listing contains a large number of corrections, problem-specific alternatives,

---

[1] T. Haigh, M. Priestley, C. Rope, "Los Alamos Bets on ENIAC: Nuclear Monte Carlo Simulations, 1947-8", *IEEE Annals of the History of Computing*, 36:3 (July-September 2014):42–63. T. Haigh, M. Priestley, C. Rope,, *ENIAC In Action: Making and Remaking the Modern Computer* (MIT Press, 2016).
[2] Undated flow diagram on one page, letter "P" and "S" circled at top left. John von Neumann papers, Library of Congress, Washington, DC., box 11, folder 7. (This collection cited hereafter as JvN-LoC.)
[3] Undated flow diagram, reversed white on black image. JvN-LoC, box 11, folder 7.
[4] 30 handwritten pages, with cover sheet reading "Card Diagram, FLOW DIAGRAM, Coding, Function Table III Values, Monte Carlo Second Run": JvN-LoC, box 12, folder 5.
[5] *Actual Running of the Monte Carlo Problems on the ENIAC*, 25 page undated, anonymous typescript. JvN-LoC box 12, folder 6.
[6] Flow diagram headed "MONTE CARLO" and dated "12/7/1947": JvN-LoC, box 11, folder 8.
[7] Undated, unheaded flow diagram, JvN-LoC, box 11, folder 8.We believe this to be a diagram for the Monte Carlo "Third Run", and it is referred to by this name in the remainder of this report.

and deleted code. We have extracted from this material the code corresponding to the basic form of the program and the special process. We have also reconstructed the card layout used and the usage of the ENIAC's accumulators and function tables in the Second Run. We have created a flow diagram which matches as closely as possible the reconstructed code. For the most part, this is identical with the Second Run flow diagram itself, but it also includes a description of the special process.

2 Description of the Second Run code listing

The code listing consists of a cover sheet and 29 pages of ruled paper divided into 6 columns. The first column contains the code for the Second Run program, giving both numeric and mnemonic forms of each instruction.

Annotations in the far left-hand margin of the code listing establish cross-referencing between the code and the Second Run flow diagram. Circled letters correspond to remote connections on the flowchart, and numbers preceded by a # symbol denote the numbered operation boxes. Extended braces show which instructions in the code belong to which operation box. The code is divided up into sequentially numbered six-line sections, each corresponding to a line in one of ENIAC's function tables where the instructions were stored. Numbers immediately to the left of the code at the head of each six-line block identify the line in which the instructions were placed. In the remainder of this report, terms of the form "FT142" or simply "142" are used to refer to a particular line in a function table, and terms of the form "FT142.5" or "142.5" to a particular operation code within a line.

The remaining five columns contain a mixture of informal textual notes and numbers. These columns are principally used to track by hand the effect of certain groups of instructions on the contents of certain accumulators. This is mostly done during tricky passages of code, presumably as a way of checking the correctness of the coding. There is no consistent correlation across pages between column and accumulator, nor is there any attempt to follow the progress of a complete run. In most cases, however, context allows us to determine which accumulators are being referred to, and the annotations are a useful guide to the precise format of the numbers stored and allow us to check our understanding of the effect of the code.

A number of corrections have been made to the code at various times: in some places codes have been rubbed out and new ones inserted, in others codes are crossed out and new ones inserted, and in yet others new codes are written in red. The correspondence between the mnemonic symbol and the code number is usually, though not always, consistent. In the reconstruction, we have not attempted to reproduce these changes, but instead have produced a "final" version incorporating this material. A list of minor editorial changes to the code is provided below.

The final page of the listing is headed "Prob. II," and is in a different format, using only five columns headed B, C, D, E and G. This page appears to be a sketch of typical data to be placed on a so-called "virgin" punched card, and read into registers B, C, D, E and G of the constant transmitter.

3 Overview of code

The Second Run flow diagram describes the computation using 84 operation and decision boxes. These are numbered sequentially, with some omissions, interpolations using diacritical marks, and a small number of unnumbered boxes.

This basic sequence of boxes seems to have been coded directly: the code sequence running from page 1 to page 22 of the listing (there are two different pages numbered "7") uses, with a few minor differences, the same numbering scheme.

Additional material in the code listing is not reflected in the flow diagram. This falls into a number of categories:

- The coding at the very beginning of the program diverges from the structure of the flow diagram. The modifications are given on "Page 1a" of the listing which appears to have been intended to precede the original page 1.
- Alterations made to the code have left gaps at some places, and also led to expansion of other sections. In the latter cases, the added code has been placed in isolated lines at non-contiguous places in the function tables, connected to the main flow with unconditional transfer instructions. Presumably this was simpler and more economical than trying to incorporate the changes inline, with the consequence of having to reorder and renumber all the existing coding.
- Pages 23 – 25 of the listing contain a material relating to the "special process" that is not described on the Second Run flow diagram or in "Actual Running". However, the structure of this section, and its place in the overall computation, is similar to that described in boxes 49, 50, 51, 65 and 66 of the Third Run flow diagram, and a significant part is annotated as belonging to box 52*, a designation that appears to relate to the box in the December 1947 First Run flow diagram corresponding to the Third Run's box 66. These materials have been used to complete the annotation of this section of the Second Run code.
- The special process code has variants and additions that are noted as applying specifically to Problem IV. Some of these have been crossed out, including a significant section on the unnumbered page at the end of the listing, FT006 – FT010. This material has not been analyzed.

It is natural to wonder if the team produced a flow diagram that reflected the final version of the code more closely than the Second Run diagram. "Actual Running" confirms that the Special Process section was not included in the diagram, so its absence does not provide an argument for a later version. On the other hand, the diagram does not look like a finished product, although its high degree of consistency with the main run of the code suggests that the code was developed from it, or something very like it. No evidence has come to light suggesting that later modifications to the code were reflected in the design documentation, however.

4 High-level organization of the program

The program code and the problem-specific data were placed in ENIAC's three function tables, here referred to as FT1, FT2 and FT3. FT1 and FT2 held the program code and FT3 held the numerical data that would be set up for each problem within the run. The contents of the function tables in the Second Run program are detailed in the table at the end of this section.

Each function table contained 104 rows, numbered −2 to 101 as shown in the leftmost column of the table. The program uses the logical addresses 000 – 099, 100 – 199 and 500 – 599 to refer 100 rows in tables FT1, FT2 and FT3, respectively. The way in which these addresses mapped onto the 104 rows in the function tables is shown for each function table in the "address" columns.

FT1 and FT2 held the program code. The table lists the two-digit codes found in the addressable rows of FT1 and FT2. These digits would have been physically set on the switches of each row of the function tables. Empty positions which should never have been accessed by the program were set to 00 (operation code 00 is a halt instruction). Some of these are shown as 00 in the code, but others are left blank. We have left them all blank in the table to make clear which areas of the function tables were used and which unused. All occurrences of 00 in the table are therefore inline data (a numeric constant, or part of a function table address), rather than operation codes. In this document we use the logical address to refer to function table rows. The rows in FT1 and FT2 are divided into six two-digit fields, which we identify using a decimal notation: 148.3, for example, refers to the field comprising digits 5 and 6 in row (physical) row 46 of FT1.

To enable easy cross-referencing, the table also shows the region codes used in our analysis of the program. Horizontal lines separate the code belonging to different regions. It is notable that not all regions correspond to contiguous blocks of code. We believe that this is because of modifications made to the code after it was first written: it would be easier to place new code in an unused part of the table than to relocate all the instructions following an insertion. As a result, the clear region by region structuring of the code is compromised at times by some rather ad hoc "spaghetti coding".

The table also indicates where the code corresponding to each operation box begins and ends. Small vertical lines mark the boundaries of operation boxes where these appear within a row.

FT3 held the constant numerical data required for each problem within the run. The same layout was used throughout, though some of the numerical values stored would vary from problem to problem. The twelve digits in a row were read as two six-digit numbers, A and B. For the most part, two quantities were held in each row, one each in the A and B sections. In the table, this data is denoted using the mathematical variables that are used in "Actual Running". Most of the data consists of a number of one and two dimensional arrays. The base address of each array is given in the code, and then incremented according to the actual value being accessed. The numbers at 590_A – 593_A are constants used in one particular section of the program.

The main program is shown in black, and the code relating to the special process and the problem IV additions is shown in blue. Codes in red are editorial corrections, discussed in detail below. Codes (or in some cases variables) in bold italic letters are parameters which could be set by hand before a program run started. These are summarized below. Highlighted entries are editorial corrections to the code, discussed in the detailed section-by-section analysis that follows.

4.1 Basic program structure

The basic structure of the Second Run program was shown in the Second Run flowchart. This corresponded to the code that would be run as part of every problem, and is shown in the table in black.

4.2 The special process

There were some minor variations made to the program for some problems, but the major variation was known as the "special process". This consisted of some additional code in region H, stored in FT2 at addresses 190 – 194, and code corresponding to box 52* in the First Run flow diagram, stored in FT2 at addresses 178 – 188. If the special process was to be invoked in a particular problem, the address at 153.3-4 had to be changed to 190 in place of 154.

The special process and box 52* were not shown on the Second Run flow diagram. We have added them in our reconstructed flow diagram.

4.3 Problem-specific variants

In addition to the special process, the code listing indicates a number of minor variations for individual problems, summarized here.

4.3.1 Problem 1

At 050.1 a note in the listing reads "Changed to 00 in Problem 1". This would have the effect of stopping the execution of the program at this point, which is reached if and only if the neutron is moving in towards the center of the assembly. We do not know what specific features of problem 1 led to this change.

4.3.2 Problem 2

At 058.4-6, a note in the listing reads "Changed to N3D in Prob 2" and the alternative code is listed as "N3D 00 66". This would mean that the assignment of 1 to accumulator 5 in box 31 was not carried out, and that instead the program jumped to address 066, at the beginning of region A_2. We do not know what specific features of problem 2 led to this change.

A single page following the code listing is headed "Prob II" and appears to give the numeric values held on a virgin card, read by region A_0.

4.3.3 Problem 4

The special process appears to have a variant specifically for problem 4. This consists of the insertion of three new instructions at 192.6, 194.3, and 198.1. These insertions necessitated extending the code for the special process into row 198 in FT2. The table below shows this, but not the insertions at 192.6 and 194.3. These changes appear to be connected with the alternative code given in the listing for row 153, where two instructions are added, but confusingly, this alternative code appears in a column headed "Problem V".

The code for box 52* also has variants specific for problem IV, appearing in the listing at row 188 and 006-010. This code has been crossed out, for reasons unknown. We have shown the crossed-out code in 006-010 in the table below, to indicate the general usage of FT1 and FT2, but have not attempted to analyze it.

4.3.4 Problem 5

As indicated above, a column headed "Problem V" on page 19 of the listing contains two additional instructions in row 153 before invoking the special process. These instructions store the contents of accumulator 15 in accumulator 8, and load the contents of accumulator 9 into 15 before calling the special process. However, the only place in the code where data is stored in accumulator 9 is in the variant code for problem 4 at 194.3. Our hypothesis is therefore that these two changes belong together, and that there is an error in labelling the changes in the listing.

4.4 Hand configuration

The following aspects of the program had to be configured before running it. In the table, these are indicated by showing the affected digits in bold italics.

- 000.2–4: a five-digit number, the initial value of the card serial number $\overline{n}$.
- 015.3: depending on the type of card being read, set the address (16 or 26) of the appropriate read section (A_0 or A_1).
- 110.3–4: address for "weighted/unweighted" discrimination in zonal escapes.
- 111.5: I, the maximum zone number in this problem.
- 124.5: the address (17 or 26) of the connectors I or A to return to either A_0 or A_1.
- 153.3–4: set to "01 54" normally, or "01 90" to invoke the special process.
- 176.4: the N here refers to the special count instruction used to stop the program after 3000 repetitions of the random number subroutine. We do not know the actual code used for this instruction, which would have to be inserted before running the program.
- 195.6: an annotation here suggests that operation 19 should be replaced by 99 "for source cards". We are not sure when or why this change would be applied.

Where these changes affect the flow of control in the program, we have adopted the notation used on the Third Run flow diagram and shown them in the redrawn flow diagram as a path bifurcation at a small unmarked circle. These occur following boxes 1, 57, 62 and 69.

Table 1 : Contents of Function Tables 1, 2 and 3

	FT1				FT2				FT3		
row	region	box #	address		region	box #	address		address	A	B
-2	S	1	000	74 0$\bar{n}$ $\bar{nn}$ $\bar{nn}$ 58 62	E	55°	100	70 14 72 06 38 22			
-1			001	17 12 83 00 14		55°/56'	101	60 32 62 02 21 76			
0			002			56'	102	15 62 80 74 11 80			
1			003			56'	103	80 71 01 83 00 93			
2			004		F	55	104	72 06 41 04 22 66	500	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{00}$	$10^{-4}\sigma_{00}$
3			005			55	105	15 62 84 11 61 06	501	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{01}$	$10^{-4}\sigma_{01}$
4	~~S.P.IV~~	~~IV~~	006	~~68 41 84 00 90 07~~		55/55*	106	24 69 84 14 61 07	502	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{02}$	$10^{-4}\sigma_{02}$
5		~~IV~~	007	~~56 69 72 56 99 99~~		55*/56	107	72 05 41 62 69 72	503	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{03}$	$10^{-4}\sigma_{03}$
6		~~IV~~	008	~~83 01 89 00 00 00~~		56	108	99 04 39 66 42 62	504	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{04}$	$10^{-4}\sigma_{04}$
7		~~IV~~	009	~~56 18 72 56 99 99~~		56	109	24 38 39 60 32 62	505	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{05}$	$10^{-4}\sigma_{05}$
8		~~IV~~	010	~~83 01 89~~		56/57	110	19 62 84 ***13*** ***3*** 1 11	506	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{06}$	$10^{-4}\sigma_{06}$
9			011			57	111	66 62 41 72 ***I*** 69	507	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{07}$	$10^{-4}\sigma_{07}$
10			012		M	58	112	55 15 31 60 53 70	508	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{08}$	$10^{-4}\sigma_{08}$
11			013			58/58*	113	39 19 72 06 41 04	509	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{09}$	$10^{-4}\sigma_{09}$
12	S	1	014	56 15 31 66 62 16		58*	114	50 66 84 12 71 15	510	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{10}$	$10^{-4}\sigma_{10}$
13			015	83 00 ***16***		58*	115	24 62 69 83 01 18	511	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{11}$	$10^{-4}\sigma_{11}$
14	A_0	0	016	44 15 15 15 15 15		64	116	72 02 49 62 22 02	512	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{12}$	$10^{-4}\sigma_{12}$
15		2	017	15 22 32 80 54 57		64	117	83 01 12	513	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{13}$	$10^{-4}\sigma_{13}$
16		2/3	018	71 51 01 72 10 22		59	118	68 53 72 05 32 58	514	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{14}$	$10^{-4}\sigma_{14}$
17		3/4	019	02 31 84 02 40 20		59	119	54 15 31 52 62 18	515	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{15}$	$10^{-4}\sigma_{15}$
18		4/5	020	21 69 56 19 20 75		59/61	120	45 15 15 37 52 72	516	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{16}$	$10^{-4}\sigma_{16}$
19		5/6	021	05 67 47 14 36 66		61/62	121	01 80 17 72 02 41	517	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{17}$	$10^{-4}\sigma_{17}$
20		6	022	15 62 73 05 46 08		62	122	38 15 62 04 22 38	518	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{18}$	$10^{-4}\sigma_{18}$
21		6	023	47 18 83 00 56		62	123	15 62 84 12 51 24	519	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{19}$	$10^{-4}\sigma_{19}$
22		7	024	02 83 00 16		62	124	24 69 83 00 ***17***	520	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{20}$	$10^{-4}\sigma_{20}$
23			025			63	125	72 01 41 22 02 83	521	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{21}$	$10^{-4}\sigma_{21}$
24	A_1	10	026	44 50 84 02 60 27		63	126	00 29	522	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{22}$	$10^{-4}\sigma_{22}$
25		10	027	60 66 15 62 41 69		58°	127	72 07 84 11 81 28	523	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{23}$	$10^{-4}\sigma_{23}$
26		10	028	62 02 19 15 15 15		58°/58'	128	41 62 69 55 15 31	524	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{24}$	$10^{-4}\sigma_{24}$
27		11	029	50 66 84 03 90 30		58'	129	66 81 81 43 04 62	525	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{25}$	$10^{-4}\sigma_{25}$
28		11	030	15 72 06 41 62 69		58'	130	49 24 62 04 39 66	526	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{26}$	$10^{-4}\sigma_{26}$
29		12	031	55 52 80 20 36 66		58'	131	62 71 48 62 24 19	527	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{27}$	$10^{-4}\sigma_{27}$
30		12	032	52 42 60 72 05 71		58'	132	83 01 18	528	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{28}$	$10^{-4}\sigma_{28}$
31		12	033	11 40 40 57 40 20	N	79	133	72 31 49 22 62 02	529	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{29}$	$10^{-4}\sigma_{29}$
32		12	034	36 66 15 83 01 95		80	134	39 42 07 27 41 05	530	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{00}$	$10^{-4}\sigma_{30}$
33		12/13	035	18 73 05 60 80 08		80	135	55 15 84 11 21 36	531	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{31}$	$10^{-4}\sigma_{31}$
34		16	036	55 15 31 52 53 38		80	136	31 60 77 42 25 69	532	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{32}$	$10^{-4}\sigma_{32}$
35		16	037	62 39 19 50 15 31		81	137	62 48 62 83 01 69	533	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{33}$	$10^{-4}\sigma_{33}$
36		16	038	01 83 00 42		82	138	39 60 42 04 72 02	534	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{34}$	$10^{-4}\sigma_{34}$
37		14	039	54 18 55 20 31 80		82	139	41 24 84 14 21 40	535	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{35}$	$10^{-4}\sigma_{35}$
38		14/15	040	81 66 62 19 73 06		82/83	140	69 72 32 49 22 62	536	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{36}$	$10^{-4}\sigma_{36}$
39		15	041	60 80 08 83 00 36		83	141	02 83 01 34	537	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{37}$	$10^{-4}\sigma_{37}$
40	C	21	042	40 86 11 62 80 05		84	142	36 70 66 15 72 05	538	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{38}$	$10^{-4}\sigma_{38}$
41		21	043	25 57 14 31 41 25		84	143	41 62 84 13 31 44	539	$(\sigma_i/\sigma_t, \sigma_E/\sigma_t)_{39}$	$10^{-4}\sigma_{39}$
42		21/22	044	57 34 41 14 62 41		84/85	144	69 72 02 38 22 60	540		$-V_5$
43		22	045	07 62 84 05 10 46		85	145	32 62 02 83 01 34	541	$-V_1$	$-V_9$
44		22/23	046	69 39 66 15 62 75	H	65	146	99 99 84 19 71 71	542	$-V_2$	$-V_8$
45		23	047	05 65 28 08 47 11		65	147	62 53 53 73 05 00	543	$-V_3$	$-V_7$
46		23	048	31 57 84 05 10 49		65	148	08 36 81 76 04 62	544	$-V_4$	$-V_6$
47		23/24	049	34 12 62 41 69 62		65/66	149	41 10 47 15 31 76	545	$-V_5$	$-V_5$
48		24	050	64 41 83 00 55		66	150	05 62 84 15 81 51	546	v_{00}	$v(\xi)_0$
49		25	051	39 38 62 32 72 02		66/67/68	151	30 10 30 69 25 52	547	v_{01}	$v(\xi)_1$
50		25	052	77 62 19 39 66 15		68/69	152	43 30 84 16 41 53	548	v_{02}	$v(\xi)_2$
51		25	053	62 75 05 66 28 08		69	153	69 83 ***01 54***	549	v_{03}	$v(\xi)_3$
52		25	054	47 11 31 57 34 64	L	70	154	72 45 28 08 47 15	550	v_{04}	$v(\xi)_4$
53		26	055	27 14 28 42 42 78		70/71/72	155	31 66 84 15 71 56	551	v_{05}	$v(\xi)_5$
54	B	30	056	68 84 06 20 57		72/73	156	41 24 69 15 72 01	552	v_{06}	$v(\xi)_6$
55		30/31	057	75 05 40 47 69 39		73	157	62 49 62 83 01 70	553	v_{07}	$v(\xi)_7$
56		31	058	43 71 19 72 01 05	JK	74	158	72 66 59 62 08 15	554	v_{08}	$v(\xi)_8$
57		32	059	47 15 84 06 10 60		75	159	40 52 80 20 40 40	555	v_{09}	$v(\xi)_9$
58		32	060	31 68 69 83 00 66		75	160	11 24 57 52 40 20	556	v_{10}	$c(\xi)_0$
59		33	061	25 39 19 83 00 59		76	161	72 01 37 17 39 58	557	v_{11}	$c(\xi)_1$
60		34	062	39 43 71 72 09 19		76	162	62 81 66 62 19 83	558	v_{12}	$c(\xi)_2$
61		34	063	72 01 84 06 60 64		76	163	00 42	559	v_{13}	$c(\xi)_3$
62		34/35/36	064	41 05 47 68 69 15		77	164	24 71 53 53 46 72	560	v_{14}	$c(\xi)_4$
63		36	065	25 39 19 83 00 63		77	165	05 80 11 31 31 31	561	v_{15}	$c(\xi)_5$
64	A_2		066	84 06 71 71		77	166	31 31 64 60 11 68	562	v_{16}	$c(\xi)_6$
65		40	067	36 60 72 05 80 43		77/78	167	57 18 72 56 59 62	563	v_{17}	$c(\xi)_7$
66		40	068	03 04 75 05 90 47		78	168	08 83 01 59	564	v_{18}	$c(\xi)_8$
67		40	069	15 31 10 47 15 15	N	81	169	27 39 19 83 01 13	565	v_{19}	$c(\xi)_9$
68		42	070	23 30 84 07 30 71	L	73	170	22 02 83 01 12	566	v_{20}	r_0
69		42/43	071	69 23 23 03 31 24	G		171	15 36 38 62 86 11	567	v_{21}	r_1
70		43	072	04 83 00 70			172	62 10 15 15 15 15	568	v_{22}	r_2
71		45	073	47 15 23 31 07 23			173	62 80 57 07 31 57	569	v_{23}	r_3
72		45	074	41 31 63 03 31 60			174	52 27 27 07 30 80	570	v_{24}	r_4
73		45	075	10 23 11 31 32 57			175	11 31 57 80 27 46	571	v_{25}	r_5
74		45	076	07 47 15 27 27 31			176	66 62 16 ***N*** 69	572	v_{26}	r_6
75		45	077	11 27 57 30 11 23			177		573	v_{27}	r_7
76		45/46	078	23 57 24 42 04 39	JK	52*	178	24 66 10 62 04 15	574	v_{28}	r_8
77	D	46	079	66 15 73 05 76 62		52*	179	15 40 86 11 62 80	575	v_{29}	r_9
78		46	080	08 47 42 39 19 39		52*	180	20 31 41 40 03 23	576	v_{30}	b_0
79		46	081	49 15 62 77 62 75		52*	181	57 13 40 57 11 30	577	v_{31}	b_1
80		46	082	05 00 28 08 47 10		52*	182	57 11 30 41 46 57	578	v_{32}	b_2
81	E	47	083	34 11 84 08 70 84		52*	183	64 14 30 64 43 07	579	v_{33}	b_3
82		47/48	084	30 57 41 24 69 30		52*	184	73 05 56 24 08 47	580	v_{34}	b_4
83		48	085	32 07 24 32 63 14		52*	185	11 34 57 13 30 11	581	v_{35}	b_5
84		48	086	72 05 38 83 00 88		52*	186	23 57 43 63 41 40	582	v_{36}	b_6
85		49	087	72 04 38 15 15 15		52*	187	52 40 20 27 71 11	583	v_{37}	b_7
86		50	088	22 71 43 62 02 21		52*	188	68 83 01 67	584	v_{38}	b_8
87		50	089	70 42 84 09 90 90			189		585	v_{39}	b_9
88		50	090	11 68 57 12 34 42	H	S.P	190	39 66 84 15 41 91	586		
89		50/53	091	62 69 68 32 07 34		S.P	191	15 72 02 41 62 69	587		$(\sigma_{2H}/\sigma_t)_{00}$
90		53	092	53 63 21 01 15 15		S.P	192	72 86 28 08 47 30	588		$(\sigma_{2H}/\sigma_t)_{01}$
91		54	093	40 80 41 40 07 34		S.P	193	10 30 84 17 81 94	589		$(\sigma_{2H}/\sigma_t)_{02}$
92		54	094	11 34 57 13 27 27		S.P	194	69 83 01 54	590	−0.005	$(\sigma_{2H}/\sigma_t)_{03}$
93		54	095	57 13 40 11 40 57	A1	12	195	62 73 05 46 08 ***19***	591	α	$(\sigma_{2H}/\sigma_t)_{04}$
94		54	096	64 53 72 05 32 80		12	196	47 83 00 35	592	010000	$(\sigma_{2H}/\sigma_t)_{05}$
95		54	097	20 27 34 41 40 52	H	65	197	39 49 15 83 01 47	593	1/3	$(\sigma_{2H}/\sigma_t)_{06}$
96		54	098	40 20 83 01 04		S.P.IV	198	08 83 01 54	594		$(\sigma_{2H}/\sigma_t)_{07}$
97		55°	099	62 41 83 01 00			199		595		$(\sigma_{2H}/\sigma_t)_{08}$
98									596		$(\sigma_{2H}/\sigma_t)_{09}$
99									597		
100									598		
101									599		

5 Input and output

Table 2, at the end of this section, shows the data that is read and printed by the program. Input data is read from punched cards and stored in the constant transmitter. Two input card formats are used: "virgin cards" contain timing information allowing the program to generate new neutrons, while "normal cards" contain detailed data about neutron properties.

Data is moved from the constant transmitter to an accumulator when needed in the processing. At the end of processing, a new punched card is printed with the data for the successor neutron, taken from accumulators 1, 2, 15, 16, 17, 18, 19 and 20, whose correlation with the 80 columns of the punched card is shown in Table 2. So, for example, the value of the neutron's velocity is stored in columns 51-55 of a normal punched card, and is moved from there, via register F of the constant transmitter, into the left-hand side of accumulator 18. Throughout the calculation, this accumulator is used to hold the velocity, including any updated value, and it is then printed back out to columns 51-55 of the output card at the end of the computation.

Reading from the left, the columns of Table 2 give the following information:

- "Columns" gives the column numbers of the fields on the punched cards.
- "CT register" identifies the register on the ENIAC's constant transmitter that the card data is read into.
- Four columns describe the contents of the "virgin" or "source" cards. The first two columns show the mathematical notation used on the flow diagram to denote this data, and an informal description of it. The column headed "Problem 2" shows the data punched on a virgin card in Problem 2 (see above). The apparently inconsistent signs are discussed in the accompanying *Computation Analysis* document. The "When read" column refers to the operation box on the flow diagram in which this data is read from the constant transmitter into the temporary storage locations provided by accumulators 11 and 15.
- The next four columns describe the contents of a "normal" card. The "Where stored" column lists the accumulator locations where the input data was moved to for processing.
- The "Accumulator" column lists the accumulator subfields from which data was read when a card was printed.
- "Print" lists the contents of these accumulators at the time of printing.
- "Description" explains the meaning of the variables listed in the "Print" column.

The last two rows of Table 2 show the values entered by hand on registers J and K of the constant transmitter before the start of a run.

- Constant transmitter J: the initial random number, ξ_0. On page 1a of the code listing, this is given as "P5679009876".
- Constant transmitter K: "1000000200", initial values of i and ε for virgin cards.

Table 2 : Input and Output

Columns	CT register	Virgin card				Normal card				Accumulator	Print	Description
			Description	Problem 2	When read		Description	When read	Where stored			
1 – 7	$\mathbf{A_{10\text{-}4}}$					t	Time	# 16	[1]	$\mathbf{[1]_{10-4}}$	t*	Time
8 – 10	$\mathbf{A_{321}}$									$\mathbf{[1]_{3-1}}$		
11	$\mathbf{B_{10}}$					α	Event type	# 11, # 58*		$\mathbf{[2]_{10}}$	α*	Event type
12	$\mathbf{B_9}$					q	Card weight	# 10	$[2]_1$	$\mathbf{[2]_9}$	q*	Card weight
13 – 15	$\mathbf{B_{8\text{-}6}}$									$\mathbf{[2]_{8-6}}$		
16 – 19	$\mathbf{B_{5\text{-}2}}$									$\mathbf{[2]_{5-2}}$	$\underline{n}$	Virgin card iteration
20	$\mathbf{B_1}$									$\mathbf{[2]_1}$	q	Parent weight
21 – 30	**C**	$-10^{-3}\,T_1$	End time	010.8172	# 2					**[15]**		
31 – 40	**D**	$+10^{-3}\,T_0$	Start time	−010.8192	# 2					**[16]**	ξ*	Last random number
41 – 45	$\mathbf{E_L}$					$\bar{n}$	Serial no.	# 59	$[18]_R$	$\mathbf{[17]_{10-6}}$	$\bar{n}$	Serial no.
46 – 50	$\mathbf{E_R}$	γ	Time interval	.0001	# 2					$\mathbf{[17]_{5-1}}$	s	Scatterings
51 – 55	$\mathbf{F_L}$					v	Velocity	# 14	$[18]_L$	$\mathbf{[18]_{10-6}}$	v*	Velocity
56 – 60	$\mathbf{F_R}$									$\mathbf{[18]_{5-1}}$	n'	Parent card
61	$\mathbf{G_{10}}$					i	Zone	# 16	$[19]_{10}$	$\mathbf{[19]_{10}}$	i*	Zone
62	$\mathbf{G_9}$			1		i_l	Last census zone	# 58	$[19]_9$	$\mathbf{[19]_9}$	i_l	Last census zone
63	$\mathbf{G_8}$					k_l	Last census velocity	# 58	$[19]_8$	$\mathbf{[19]_8}$	k_l	Last census velocity
64	$\mathbf{G_7}$			1		i_m	Furthest zone	# 58, # 80	$[19]_7$	$\mathbf{[19]_7}$	i_m*	Furthest zone
65 – 69	$\mathbf{G_{6-2}}$									$\mathbf{[19]_{6-2}}$		
70	$\mathbf{G_1}$					k	Velocity interval	# 58'	$[19]_1$	$\mathbf{[19]_1}$	k*	Velocity interval
71 – 75	$\mathbf{H_L}$					r	Distance	# 12, # 16	$[20]_L$	$\mathbf{[20]_{10-6}}$	r*	Distance
76 – 80	$\mathbf{H_R}$					a	Direction	# 16	$[20]_R$	$\mathbf{[20]_{5-1}}$	a*	Direction
	J	5879009876				ξ_0	Initial random no.					
	K	1000000200				i, ε	Initial values					

6 Accumulator usage

Table 3 shows how the data used by the program is mapped onto the available accumulators and which sections of the program processes particular data values.

The first and last columns identify the logically distinct storage areas used. Some of these correspond to complete accumulators, but in other cases a number of different data values were stored in subfields within a single accumulator.

Within the body of the table, a column headed “X” (for each region X) identifies the contents of the accumulators as the region is entered, and a column headed “X*” shows their contents as the region is exited. The difference between these two columns therefore summarizes the effect of the code in that region.

Some regions make use of otherwise unused accumulators as temporary storage space which was written to and read within the region, but then overwritten or ignored once execution of that region was complete. The table does not document these “local variables”.

The column headed “Print” shows the values stored at the point in region M when a new card is printed.

Table 3 : Accumulator usage by region

Acc	S*	A_0	A_0*	A_1	A_1*	C	C*	B	B*	A_2	A_2*	D	D*	E	E*	F	F*	M	Print	M*	N	N*	H	H*	L	L*	JK	JK*	Acc
$1_{10\text{-}4}$			t		t	t	t	t	t	t	t	t	t	t	t*	t*	t*	t*	t*	t*	t*	t*	t*	t*	t*	t*	t*	t*	$1_{10\text{-}4}$
1_{321}																													1_{321}
2_{10}															α*	α*	α*	α*	α*	α*	α*	α*	α*	α*	α*	α*	α*	α*	2_{10}
2_9																		0 or q*	q*	q*		q*				q*			2_9
$2_{8\text{–}6}$																													$2_{8\text{–}6}$
$2_{5\text{–}2}$	0 → n	n	n+1→n					(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	$2_{5\text{–}2}$
2_1			0 → q		q	q	q	q	q	q	q	q	q	q	q	q	q	q	q	q − 1	q	q	q	q	q	q	q		2_1
3																													3
4											$10^{-4}\lambda$	$10^{-4}\lambda$	$10^{-4}\lambda$	$10^{-4}\lambda$										ξ_{321}			ξ_{321}		4
5																													5
6																													6
7															$10^{-5}\sigma$														7
8_{10}																													8_{10}
8_{987}					θ	θ																							8_{987}
$8_{6\text{-}1}$																													$8_{6\text{-}1}$
9																													9
10													$10^{-4}\sigma$	$10^{-4}\sigma$															10
11																													11
12																													12
13																													13
14			d				d	d	d	d	d	d	d	d	d_2	d_2	d_2	d_2		d_2	d_2	d_2	d_2	d_2	d_2	d_2	d_2	d_2	14
15																													15
16	ξ	ξ	ξ	ξ	ξ	ξ	ξ	ξ	ξ	ξ	ξ*	ξ*	ξ*	ξ*	ξ*				ξ'				ξ**	ξ**	ξ**	ξ**	ξ**		16
17_L	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄	n̄			n̄	n̄	n̄+1→n̄									17_L
17_R			0 → s		0 → s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	s	s	s	s	s	s	s	s+1→s	17_R
18_L			v		v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v* → v	18_L
18_R																			n'	n'									18_R
19_{10}			1 → i	(i*)	i	i	i	i	i	i	i	i	i	i	i	i	i	il or i*	i*	i*	i*	i*	i*	i	i	i	i	i	i	19_{10}
19_9				(i_l*)															i_l*	i_l*									19_9
19_8				(k_l*)															k_l*	k_l*									19_8
19_7				(i_m*)														(i_m)	i_m*	i_m*		(i_m)							19_7
19_{654}																													19_{654}
19_3			2 → ε	(ε)			ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	ε	(ε)	ε	ε	ε	ε	ε	ε	ε	ε	0	19_3
19_2				(b')								b	b	b	b	b'	b'	(b')	b'	b'	b'	b'	b'	b'	b'	b'	b'	0	19_2
19_1				(k)	(k)	(k)	(k)		k	k	k	k	k	k	k	k	k	k	k	k	k	k	k	k	k	k	k	k	19_1
20_L			0 → r		r	r	r	r	r	r	r	r	r	r	r*	r*	r*	r*	r*	r*	r*	r*	r*	r*	r*	r*	r*	r* → r	20_L
20_R			0 → a		a	a	a	a	a	a	a	a	a	a	a*	a*	a*	a*	a*	a*	a*	a*	a*	a*	a*	a*	a*	a** → a	20_R
Acc	S*	A_0	A_0*	A_1	A_1*	C	C*	B	B*	A_2	A_2*	D	D*	E	E*	F	F*	M	Print	M*	N	N*	H	H*	L	L*	JK	JK*	Acc

7 Reconstruction of code and flow diagram

In this section we present an annotated listing of the code in each region of the Second Run program. The functionality of each region and its role in the overall computation are discussed in the accompanying *Computational Analysis*. We have also produced a flow diagram which accurately documents the functionality of the code in the spreadsheet. It is a revised version of the archival Second Run flow diagram that incorporates the following changes.

- The Second Run diagram appears to be a draft, and it is rather hard to read in some places. We have altered the layout of the diagram where this seemed to aid readability.
- Boxes and the formulas within them are consistent with the code spreadsheet, and in places differ from those given on the archival diagram.
- The reconstructed flow diagram includes the special process which, according to "Actual Running", was missing from the Second Run flow diagram and is not shown on the archival diagram.
- In some places, the formulas in the boxes of the Second Run diagram are rather hard to read. We have occasionally made use of the "Third Run" flow diagram to clarify the contents of certain boxes.

For the most part, the code is a step by step implementation of the functionality described in Second Run flow diagram. In the comments on each region, below, we note the places where the code deviates from the diagram. These deviations are reflected in our reconstructed flow diagram. Some of these changes in turn appear in the later flow diagram, which has been used where noted to inform the presentation of the reconstructed flow diagram.

The columns on the code listings have the following significance:

- **Label:** transcription of labels, box numbers and occasionally FT line numbers showing where control comes from to get to this FT line. Labels are taken from the left margin of the code listing, but labels and FT line numbers have been generated in the course of analyzing the code. The default control mechanism is sequential flow from one line to the next, and is not explicitly shown.

- **Box #:** the numbered box corresponding to this section of code, taken from the left margin of the code listing. The boundaries of the operation boxes shown by extended braces in the code do not always agree with those on the flow diagram, and where necessary we have moved these labels to get better consistency: these moved labels are written in red.

- **FT**: the address of the FT row.

- **Symbol:** the mnemonic symbol for a single instruction. In the cases of instruction such as N3D6 which move numbers into an accumulator, the actual digits to be moved are shown in this column in the listing, but in the spreadsheet are omitted for clarity and shown only the following "code" column.

- **Code:** the operation code (or data value as a two digit number, in the case of the 1-address instructions such as N3D8 which are followed immediately by data values).

- **Effect on accumulator contents**: a description of what that instruction does. The notation used for this is described below. Instructions that occupy more than one operation slot in the FT are annotated in the final position.

- **Formula from flowchart boxes**: a formula from the flow diagram, placed at the point at which the code *completes* the calculation specified by the formula. Transfers are shown using an IF … → notation: the expression following the IF is taken from the conditional box on the flow diagram, and the destination of the jump is shown by a "flowchart address" after the →.

To describe the effect of each instruction, the following notation is used. It is derived from notation used in the flow diagrams and other material describing the Monte Carlo programs, but we have standardized usage as in archival materials this notation is informal and often inconsistent.

- Function table line numbers are always written as 3 digits, eg 007, 195 etc.
- Box numbers from the flowchart are written # 1, # 42 etc.
- Square brackets are used to flag references to accumulators [1] – [20] and constant transmitter registers [A] – [K].
- The subscripts L and R are used to refer to the left and right halves of these registers, in cases where two numbers are held.
- Digit subscripts are used to refer to individual digits or short sequences of digits in accumulators. So $[15]_9$ is the 9^{th} digit in accumulator 15 (the second from the left), and $[6]_{321}$ is the rightmost three digits of accumulator 6. To avoid confusion we use commas when, and only when, the 10^{th} digit is referred to: the leftmost 3 digits of accumulator 12, for example, would be denoted $[12]_{10,9,8}$ and not $[12]_{1098}$.
- Most lines indicate the content of one or more accumulators after the execution of the operation in question, with an expression of the form [15] = 0. Semi-colons delimit these expressions where necessary.
- The data on the right hand side of these formulas can be a number, a variable or a more complex expression.
- Expressions of the form $FT(n)_A$ and $FT(n)_B$ denote the contents of left and right hand numbers, respectively, stored in row n of the numeric function table.
- On occasions, particularly where shift operations are being used, is it useful to track individual digits in an accumulator. The use of letters for this purpose, as in "[15] = abcdefghij", has been borrowed from Fritz's 1949 BRL report on the ENIAC. Examples taken from annotations in the code listing have been placed in quotes, as in: [15] = "5679009876". Repeated digits are sometimes indicated by an exponent, so "$0^9$1" is equivalent to "0000000001", for example.

The following small editorial changes have been made to the code presented in the tables below:

- Operation symbols are written with upper-case letters. For example, operation 11 is represented on the spreadsheet by 11L instead of 11l, and operation 15 by CL instead of cl.
- Operations 72, 73 and 74 have been consistently given the symbols N2D15, N4D15 and N6D15, respectively.
- At 059.3, operation code 24 has been amended to 84, consistent with the given symbol N6D6.
- At 117.1, the symbol N3D has been amended to N3D6, consistent with the given operation code 83.

7.1 Region S

On the Second Run flow diagram, the program starts at connector S which, after initializing the random variable ξ in accumulator 16, goes directly to read a virgin card. The diagram does not explain how a non-virgin card deck is read, as there is no way to get from S to region A_1.

The code provides a complete implementation. The label *S* also appears on page 1a, which precedes page 1. This page contains code in rows 000, 001 and 014 initializing the serial number $\bar{n}$ and the random number seed ξ, corresponding to box 1 on the flow diagram. The code in row 015 performs an unconditional transfer to 016, the next row. However, the digits 16 are circled and followed by a note that appears to read "or 26". As region A_0 starts at row 016, and A_1 at 026, we assume that row 015 transfers control to either region A_0 or A_1, depending on the address set by hand in 015.3 before the program starts.

The Third Run flow diagram more accurately reflects the structure of the Second Run code at this point, separating the common start-up code in rows 000, 001, 014 and 015 from the read operations and introducing a branching notation to show the effect of manually setting the transfer address. In the reconstructed flow diagram we have adopted this notation, and we have placed the start-up code in a new region, S.

Both the code and an annotation on page 1a showing the initial values of constant transmitter registers J and K indicate that ξ_0 was read from register J of the constant transmitter, not K as shown in the annotation box preceding box 1 on the Second Run flow diagram.

We have added a formula to box 1, at 001.1, recording the effect of the code in 000−001.1 which copies the initial serial number for this run from 000.2−4 to accumulator 17. We have also added the box number #1 which was omitted on page 1a of the code listing, presumably by a simple oversight.

The operator Θ in box 1 denotes rotation of the digits of ξ_0 one place to the left. This is carried out by the instructions in row 014, and the effect of these operations is traced in annotations in the code listing. The annotations state that seed value 5679009876 is read from the constant transmitter and that, after rotation, the value 679009875 is stored in accumulator 10. This appears to be an error: after rotation, the value stored in accumulator 10 should be 6790098765, as shown in an annotation to (deleted) instruction 172.2 in region G, on page 22 of the listing. (At this point, the current value of ξ is being retrieved for use in calculating the next value, and the annotations tracing the effect of the code take the initial value as a starting point.)

Region S - Initialization

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
(S) →	# 1	000	N6D15	74		
				$0\overline{n}$		
				$\overline{nn}$		
				$\overline{nn}$	[15] = $0^5\overline{nnnnn}$	
			S'R5	58	[12] = $\overline{nnnnn}0^5$; [15] = 0	
			12T	62	[15] = $\overline{nnnnn}0^5$	
		001	17L	17	[17] = $\overline{nnnnn}0^5$; [15] = 0	$\overline{n}$ → **[17]$_L$**
			12L	12	[12] = 0	
			N3D6	83		
				00		
				14	→ 014	
		014	JK	56	[11] = [J] ; [15] = [K]	**ξ_0 in [J]**
			CL	15	[15] = 0	
			11T	31	[15] = [J] = "5679009876"	
			S'L1	66	$[12]_1$ = "5" ; [15] = "6790098760"	
			12T	62	[15] = "679009876**5**"	**$\xi_0 = \Theta\xi_{\_}$; $\xi_0 \rightarrow \xi_n$**
			16L	16	[16] = "679009876**5**" ; [15] = 0	**$\xi_n \rightarrow$ [16]**
		015	N3D6	83		
				00		
				16	→ 016 or → 026	**→ (I) # 0 or (J) # 10**

7.2 Region A_0

The unnumbered box on the flow diagram that contains the read operation is given the number 0 in the code listing.

In box 2 of the Second Run flow diagram the variable holding the neutron's start time is given as t^*. Asterisks are used throughout the Monte Carlo source materials to denote updated values of variables, but the value calculated here is more naturally thought of as the initial value for the current cycle of the program; compare, for example, the initial value of v in box 6, which has no asterisk. In the draft diagram and in box 4 of the flow diagram t does not have an asterisk, so we have removed it in box 2.

Box 2 also asserts that the value of t is placed in accumulator [1_L]. As the value is seven digits, it occupies more than the left-hand half of the accumulator. As t is the only variables stored in accumulator 1, we have removed the reference to the left-hand half.

Box 5 of the Second Run diagram states that the initial values of i_l and i_m are set to 1. In the notes giving the contents of a virgin card for problem 2, the value 1 is shown in the positions on the input card from which the values of these variables would normally be read. However, the code does not set these variables, and so the formulas have been omitted from box 5 of the redrawn flow diagram.

The instruction SL2 (shift left by 2 places) in 018.1 is possibly incorrect. Time values are stored as negative numbers of seven digits, such as the value -010.8192 given in annotations in the code. Here 010 represents the census interval, and 8192 the time within the census interval. The code in 017 calculates the time increment $\underline{n}\gamma$ as a four digit number at the far right of accumulator 15. This would need to be shifted left by three places, not two, to align with the last four digits of the existing time value.

The code at 023.5 was corrected from 57 to 56 in the original code listing.

Region A_0 : Read a virgin card and generate new neutrons

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
(J) # 1, # 7 →	**# 0**	016	READ	44	[A] – [H] = card contents	**Read**
			CL	15		
			CL	15		
			CL	15		
			CL	15		
			CL	15		
(A) # 62 →	**# 2**	017	CL	15	[15] = 0	
			2T	22	[15] = 00000$\underline{nnnn}$0	
			SR1	32	[15] = 000000$\underline{nnnn}$	
			SL5	80	[15] = 0$\underline{nnnn}$00000 = $\underline{n}$	
			EF	54	[11] = [E] = $10^{-5}\gamma$; [15] = $\underline{n}$ + [F] = $\underline{n}$	
			×	57	[15] = $10^{-5}\underline{n}\gamma$	
		018	SL2	71	[15] = $10^{-3}\underline{n}\gamma$	
			CD	51	[11] = $-10^{-3}T_1$; [15] = $10^{-3}\underline{n}\gamma + 10^{-3}T_0$	
			1L	01	[1] = $10^{-3}(+T_0 + \underline{n}\gamma) = 10^{-3}t$; [15] = 0	**$10^{-3}t = 10^{-3}(+T_0 + \underline{n}\gamma)$; $10^{-3}t$ → [1]**
	# 3		N2D15	72		
				10	[15] = 10	
			2T	22	[15] = 00000$\underline{nnnn}$0 + 10 = $\underline{n}$ + 1	
		019	2L	02	[2] = $\underline{n}$ + 1 ; [15] = 0	**$\underline{n}$ + 1 → $\underline{n}$; $\underline{n}$ → $[2]_{5\text{-}2}$**
	# 4		11T	31	[15] = $-10^{-3}T_1$	
			N6D6	84		
				02		
				40		
				20	$[6]_{654}$ = 024 ; → 020	
		020	1T	21	[15] = $10^{-3}(t - T_1)$	
			CT	69	→ 024 if [15] ≥ 0 ; [15] = 0	**IF $t - T_1 \geq 0$ → # 7**
	# 5		JK	56	[11] = [J] ; [15] = [K] = "1000000200"	
			19L	19	[19] = "1000000200" ; [15] = 0	**$i = 1 \rightarrow [19]_{10}$; $\varepsilon = 2 \rightarrow [19]_3$**
			20L	20	[20] = 0	**0 → [20]**
			N3D8	75		
		021		05		
				67	$[8]_{321}$ = 567	
			FTN	47	[15] = $FT(567)_B = r_1$	
			14L	14	[14] = r_1 ; [15] = 0	**$d = r_1$ → [14]**
	# 6		16T	36	[15] = ξ	
			S'L1	66	$[12]_1 = \xi_{10}$	
		022	CL	15	[15] = 0	
			12T	62	$[15]_1 = \xi_{10}$	
			N4D15	73		
				05		
				46	[15] = 546 + ξ_{10}	
			8L	08	$[8]_{321}$ = 546 + ξ_{10} ; [15] = 0	
		023	FTN	47	[15] = $FT(546 + \xi_{10})_B = v(\xi_{10})$	
			18L	18	[18] = $v(\xi_{10})$ = v ; [15] = 0	**$v = v(\xi_{10})$ → [18]**
			N3D6	83		
				00		
				56	→ 056	**→ (Θ_2) # 30**
# 4 →	**# 7**	024	2L	02	[2] = 0	**0 → $\underline{n}$**
			N3D6	83		
				00		
				16	→ 016	**→ # 0**

7.3 Region A_1

The read operation is shown on the Second Run flow diagram in a separate unnumbered box. In the code listing, it is included in the section for box 10. Furthermore, this section of code has significant internal control structure: the value of q is only stored in an accumulator once a non-negative value has been read. In addition, the code clears accumulator 19 at this point. This is made explicit in the redrawn flow diagram by splitting the original box 10 into two, both labeled "10".

The path leading to box 14 is labeled "Census". In fact, cards representing a terminal event of zonal escape whose weight is not zero will also be processed in box 14, and we have added this label to the redrawn diagram.

Accumulator 19 is cleared in boxes 10 and 12 before new values are written into it. In box 12, the code for the clear operation at 195.6 was originally omitted from the code listing: the correction has been made in a marginal note, with the following 4 codes being moved one place forward in the function table. It may appear from the flow diagram that it is not cleared on the path that leads from connector K through box 14. However, the instruction at 040.4 has the effect of clearing every digit apart from the rightmost one to which k is being written.

In the code for box 12, where accumulator 19 is cleared at 195.6, an annotation in the code says "→ (99) for source cards". This suggests that in some cases the operation to clear accumulator 19 should be replaced by operation 99, a "delay" operation which had no effect. We have marked this as a configuration option to be carried out before running the program, but it is not clear to us exactly when or why this option would be chosen.

In box 16, we have removed the references to the left-hand half of register A and accumulator 1, as the variable t occupies the leftmost seven places in both. This makes boxes 2 and 16 in the redrawn diagram consistent.

We have moved the beginning of box 13 one place forward, to reflect the fact that the code at 035.1 completes the operation annotated in box 12 of the Second Run flow diagram.

Region A_1 : Read card and store neutron characteristics

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
(I) # 1, # 62 →	**# 10**	026	READ	44	[A] – [H] = card contents	**Read**
			AB	50	[11] = [A] ; [15] = [B] = αq00000000	
			N6D6	84		
				02		
				60		
				27	$[6]_{654}$ = 026 ; → 027	
		027	SL1	60	[15] = q0000000000	
			S'L1	66	$[12]_1$ = q	**q = $[B]_9$**
			CL	15	[15] = 0	
			12T	62	[15] = q	
			M	41	[15] = –q	
			CT	69	→ 026 if [15] ≥ 0 ; [15] = 0	**IF –q ≥ 0 → # 10**
		028	12T	62	$[15]_1$ = q	
			2L	02	$[2]_1$ = q ; [15] = 0	**q → $[2]_1$**
			19L	19	[19] = 0	**0 → [19]**
			CL	15		
			CL	15		
			CL	15	[15] = 0	
(K) # 63 →	**# 11**	029	AB	50	[11] = [A] ; [15] = [B] = αq00000000	
			S'L1	66	$[12]_1$ = α	
			N6D6	84		
				03		
				90		
				30	$[6]_{654}$ = 039 ; → 030	
		030	CL	15	[15] = 0	
			N2D15	72		
				06	[15] = 6	
			M	41	[15] = –6	
			12T	62	[15] = α – 6	
			CT	69	→ 039 if [15] ≥ 0 ; [15] = 0	**IF α – 6 ≥ 0 → # 14**
	# 12	031	GH	55	[11] = [G] ; [15] = [H] = (r, a)	
			SR5	52	[15] = (0, r)	
			SL5	80	[15] = (r, 0)	
			20L	20	[20] = (r, 0)	**r = $[H]_L$ → $[20]_L$**
			16T	36	[15] = ξ = abcdefghij	
			S'L1	66	[12] = a ; [15] = bcdefghij0	
		032	SR5	52	[15] = 00000bcdeg	
			SR3	42	[15] = 00000000bc	
			SL1	60	[15] = 0000000bc0	
			N2D15	72		
				05	[15] = 0000000bc5	
			SL2	71	[15] = 00000bc500	
		033	11L	11	[11] = 00000bc500 ; [15] = 0	
			20T	40	[15] = (r, 0)	
			20T	40	[15] = (2r, 0)	
			×	57	$[15]_R$ = 2r * (ξ_{32} + .005) = a	**a = 2r[ξ_{32} + .005]**
			20T	40	[15] = (r,a)	
			20L	20	[20] = (r, a) ; [15] = 0	**a → $[20]_R$**
		034	16T	36	[15] = ξ	
			S'L1	66	[12] = $0^9\xi_{10}$	
			CL	15	[15] = 0	
			N3D6	83		
				01		
				95	→ 195	

		195	12T	62	[15] = $0^{9}\xi_{10}$	
			N4D15	73		
				05		
				46	[15] = 546 + ξ_{10}	
			8L	08	$[8]_{321}$ = 546 + ξ_{10} ; [15] = 0	
			19L	***19***	[19] = 0	**0 → [19]**
		196	FTN	47	[15] = FT(546 + ξ_{10}) = v(ξ_{10})	
			N3D6	83		
				00		
				35	→ 035	
		035	18L	18	[18] = v(ξ_{10}) ; [15] = 0	**v = v(ξ_{10}) → [18]**
	# 13		N4D15	73		
				05		
				60	[15] = 0000000560	
			SL5	80	[15] = 0056000000	
			8L	08	[8] = 0056000000 ; [15] = 0	**θ_2 → θ ; θ → $[8]_{987}$**
# 15 →	**# 16**	036	GH	55	[11] = [G] = i..... ; [15] = [H]	
			CL	15	[15] = 0	
			11T	31	[15] = i.....	
			SR5	52	[15] = 00000i....	
			SR4	53	[15] = 000000000i	
			S'R1	38	[12] = i000000000 ; [15] = 0	
		037	12T	62	[15] = i000000000	
			19T	39	[15] = i00000000(k)	
			19L	19	[19] = i00000000(k) ; [15] = 0	**i = $[G]_{10}$ → $[19]_{10}$**
			AB	50	[11] = [A] = 10^{-3}t ; [15] = [B]	
			CL	15	[15] = 0	
			11T	31	[15] = 10^{-3}t	
		038	1L	01	[1] = 10^{-3}t ; [15] = 0	**10^{-3}t = [A] → [1]**
			N3D6	83		
				00		
				42	→ 042	**→ # 21**
# 11 →	**# 14**	039	EF	54	[11] = [E] ; [15] = [F] = v	
			18L	18	[18] = v ; [15] = 0	**v = [F] → 18**
			GH	55	[11] = [G] ; [15] = [H] = (r, a)	
			20L	20	[20] = (r, a) ; [15] = 0	**r = $[H]_L$ → $[20]_L$; a = $[H]_R$ → $[20\}_R$**
			11T	31	[15] = [G] = ...k	
			SL5	80	[15] = ...k00000	
		040	SL4	81	[15] = k00000000	
			S'L1	66	[12] = 000000000k ; [15] = 0	
			12T	62	[15] = 000000000k	
			19L	19	[19] = 000000000k ; [15] = 0	**k = $[G]_1$ → $[19]_1$**
	# 15		N4D15	73		
				06		
		041		60	[15] = 0000000660	
			SL5	80	[15] = 0066000000	
			8L	08	[8] = 0066000000	**θ_1 → θ ; θ → $[8]_{987}$**
			N3D6	83		
				00		
				36	→ 036	**→ # 16**

7.4 Region C

There is no code corresponding to box 20 of the Second Run flow diagram. As accumulator 19 is cleared in region A_1, this box presumably became unnecessary, and we have omitted it from the redrawn diagram.

Otherwise, the code in this region matches the Second Run flow diagram exactly. We have moved the boundaries of boxes 22 and 24 given in the code listing to reflect the facts that 044.3 completes the processing of box 21, and 049.5 performs the conditional jump at the end of box 23.

Region C : Calculate distance to zone boundary

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 16, # 76 →	**# 21**	042	20T	40	[15] = (r, a)	
			S'L5	86	$[12]_R$ = r ; $[15]_L$ = a	
			11L	11	$[11]_L$ = a ; [15] = 0	
			12T	62	$[15]_R$ = r	
			SL5	80	$[15]_L$ = r	
			5L	05	[5] = r ; [15] = 0	
		043	5T	25	$[15]_L$ = r	
			×	57	[15] = ar	
			14L	14	[14] = ar ; [15] = 0	
			11T	31	[15] = a	
			M	41	[15] = –a	
			5T	25	[15] = r – a	
		044	×	57	[15] = a(r – a) ; [12] = r – a	
			14T	34	[15] = ar + a(r – a) = Δ	**Δ = ar + a(r – a)**
			M	41	[15] = –Δ	
			14L	14	[14] = –Δ ; [15] = 0	**–Δ → [14]**
	# 22		12T	62	[15] = r – a	
			M	41	[15] = a – r	
		045	7L	07	[7] = a – r ; [15] = 0	**a – r → [7]**
			12T	62	[15] = r – a	
			N6D6	84		
				05		
				10		
				46	$[6]_{654}$ = 051 ; → 046	
		046	CT	69	→ 051 if [15] ≥ 0 ; [15] = 0	**IF r – a ≥ 0 → # 25**
	# 23		19T	39	[15] = i00000000k	
			S'L1	66	[12] = 000000000i	
			CL	15	[15] = 0	
			12T	62	[15] = 000000000i	
			N3D8	75		
		047		05		
				65	$[8]_{321}$ = 565	
			8T	28	$[15]_{321}$ = 565 + i	
			8L	08	$[8]_{321}$ = 565 + i ; [15] = 0	
			FTN	47	[15] = $FT(565 + i)_B$ = r_{i-1}	
			11L	11	[11] = r_{i-1} ; [15] = 0	
		048	11T	31	[15] = r_{i-1}	
			×	57	[15] = r_{i-1}^2	
			N6D6	84		
				05		
				10		
				49	$[6]_{654}$ = 051 ; → 049	
		049	14T	34	[15] = r_{i-1}^2 – Δ	
			12L	12	[12] = r_{i-1}^2 – Δ ; [15] = 0	**r_{i-1}^2 – Δ → [12]**
			12T	62	[15] = r_{i-1}^2 – Δ	
			M	41	[15] = Δ – r_{i-1}^2	
			CT	69	→ 051 if [15] ≥ 0 ; [15] = 0	**IF Δ – r_{i-1}^2 >= 0 → # 25**
	# 24		12T	62	[15] = r_{i-1}^2 – Δ = Δ'	**Δ' = r_{i-1}^2 – Δ**
		050	√	64	[15] = √Δ'	
			M	41	[15] = –√Δ' = L	**L = –√Δ' → [15]**
			N3D6	83		
				00		
				55	→ 055	**→ # 26**

# 22, # 23 →	**# 25**	051	19T	39	[15] = i00000000k	
			S'R1	38	$[12]_{10}$ = k ; [15] = 0i00000000	
			12T	62	[15] = ki00000000	
			SR1	32	[15] = 0ki0000000	
			N2D15	72		
				02	[15] = 0ki0000002	
		052	S'L2	77	$[12]_{21}$ = 0k ; [15] = i000000200	
			12T	62	[15] = i00000020k	
			19L	19	[19] = i00000020k ; [15] = 0	**2 → ε → $[19]_3$**
			19T	39	[15] = i00000020k	
			S'L1	66	[12] = 000000000i	
			CL	15	[15] = 0	
		053	12T	62	[15] = 000000000i	
			N3D8	75		
				05		
				66	$[8]_{321}$ = 566	
			8T	28	$[15]_{321}$ = 566 + i	
			8L	08	$[8]_{321}$ = 566 + i ; [15] = 0	
		054	FTN	47	[15] = $FT(566 + i)_B = r_i$	
			11L	11	[11] = r_i ; [15] = 0	
			11T	31	[15] = r_i	
			×	57	[15] = r_i^2	
			14T	34	[15] = $r_i^2 - \Delta = \Delta'$	**$\Delta' = r_i^2 - \Delta$**
			√	64	[15] = $\sqrt{\Delta'} = L$	**$L = \sqrt{\Delta'}$ → [15]**
# 24 →	**# 26**	055	7T	27	[15] = L + (a – r) = d	
			14L	14	[14] = d ; [15] = 0	**d = –r + a + L → [14]**
			8T	28	[15] = 0θθθxxxxxx	
			SR3	42	[15] = 0000θθθxxx	
			SR3	42	[15] = 0000000θθθ	
			6R3	78	$[6]_{321}$ = θ = 056 (θ_2) or 066 (θ_1) ; [15] = 0	**→ θ**

7.5 Region B

The Second Run flow diagram shows a connection from box 36 to box 35. However, box 35 does not start on a line boundary, and in the code the jump actually goes to 063 in the middle of box 34 and harmlessly repeats a bit of initialization from box 34 before rejoining box 35.

The flow diagram shows paths merging after boxes 32 and 35 and control passing to the subroutine. Before this can happen, the return address must be set: this is done at 066, which takes place between the merging of the paths and the connector ρ. Following the practice used in the Second Run flow diagram boxes 13 and 15, where the variable remote connection θ is set, the setting of the return address ω has been shown on the reconstructed flow diagram in an unnumbered box following boxes 32 and 35.

The boundaries of boxes 30 and 36 have been moved slightly to reflect the code.

Region B : Find neutron's velocity interval

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
(Θ_2) # 6 →	# 30	056	18T	68	[15] = v	
			N6D6	84		
				06		
				20		
				57	$[6]_{321}$ = 062 ; → 057	
		057	N3D8	75		
				05		
				40	$[8]_{321}$ = 540 = m	**m = 540 → [8]**
			FTN	47	[15] = v – V_5 ; $[8]_{321}$ = 541	**v – V_R(m) = v – V_5**
			CT	69	→ 062 if [15] >= 0 ; [15] = 0	**IF v – V_5 ≥ 0 → # 34**
	# 31		19T	39	[15] = [19] = i000000ε0k	
		058	SR2	43	[15] = 00i000000ε	
			SL2	71	[15] = i000000ε00	
			19L	19	[19] = i000000ε00 ; [15] = 0	**k = 0 → $[19]_1$**
			N2D15	72		
				01	[15] = 0000000001	
			5L	05	[5] = 000000001	**1 → [5]**
# 33 →	**# 32**	059	FTN	47	[11] = $FT(m)_A$ = –V_{k+1} ; $[8]_{321}$ = m + 1	**m + 1 → m**
			CL	15	[15] = 0	
			N6D6	84		
				06		
				10		
				60	$[6]_{654}$ = 061 ; → 060	
		060	11T	31	[15] = –V_{k+1}	**–V_L(m) → –V_{k+1}**
			18T	68	[15] = v – V_{k+1}	
			CT	69	→ 061 if [15] ≥ 0 ; [15] = 0	**IF v – V_{k+1} ≥ 0 → # 33**
			N3D6	83		
				00		
				66	→ 066	**→ 066 → ρ**
# 32 →	**# 33**	061	5T	25	[15] = 0000000001	
			19T	39	$[15]_1$ = $[19]_1$ + 1 = k + 1	
			19L	19	$[19]_1$ = k + 1 ; [15] = 0	**k + 1 → k**
			N3D6	83		
				00		
				59	→ 059	**→ # 32**
# 30 →	**# 34**	062	19T	39	[15] = i000000ε0k	
			SR2	43	[15] = 00i000000ε	
			SL2	71	[15] = i000000ε00	
			N2D15	72		
				09	[15] = i000000ε09	
			19L	19	[19] = i000000ε09 ; [15] = 0	**k = 9 → $[19]_1$**
# 36 →		063	N2D15	72		
				01	[15] = 0000000001	
			N6D6	84		
				06		
				60		
				64	$[6]_{654}$ = 066 ; → 064	
		064	M	41	[15] = –1	
			5L	05	[5] = –1 ; [15] = 0	**–1 → [5]**
	# 35		FTN	47	[15] = $FT(m)_B$ = –V_k ; $[8]_{321}$ = m + 1	**–V_R(m) → –V_k ; m + 1 → m**
			18T	68	[15] = v – V_k	
			CT	69	→ 066 if [15] ≥ 0 ; [15] = 0	**IF v – V_k ≥ 0 → 066**
	# 36		CL	15	[15] = 0	
		065	5T	25	[15] = –1	
			19T	39	$[15]_1$ = $[19]_1$ – 1 = k – 1	
			19L	19	$[19]_1$ = k – 1 ; [15] = 0	**k – 1 → k**
			N3D6	83		
				00		
				63	→ 063	**→ # 35**

7.6 Region A_2

There is no code corresponding to boxes 41 and 44 of the Second Run flow diagram, which are simply substitution boxes as shown on the First Run diagram. We have marked them as such on the reconstructed diagram.

Factors of 10^{-2} have been added to the first line in box 43 for consistency with boxes 42 and 45; this is done in the later flow diagram.

Otherwise, the code in this region matches the Second Run flow diagram exactly.

Region A_2 : Calculate random parameter λ

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
(θ_1) # 32, # 35 →		066	N6D6	84		
				06		
				71		
				71	$[6]_{654} = 067 = \omega_1$; → 171	**ω_1 → ω ; → ρ**
ω_1 →	**# 40**	067	16T	36	[15] = ξ = abcdefghij	
			SL1	60	[15] = bcdefghij0	
			N2D15	72		
				05	[15] = bcdefghij5	
			SL5	80	$[15] = ghij500000 = \xi_0$	**$\xi_0 = \xi_{4321} + 0^4 5$**
			SR2	43	$[15] = 00ghij5000 = 10^{-2}\xi_0$	
		068	3L	03	$[3] = 10^{-2}\xi_0$; [15] = 0	**$10^{-2}\xi_0$ → [3]**
			4L	04	$[4] = 0 = \alpha_0$	**$\alpha_0 = 0$ → [4]**
			N3D8	75		
				05		
				90	$[8]_{321} = 590$	
			FTN	47	$[11] = FT(590)_A = -0.005$; $[8]_{321} = 591$	
		069	CL	15	[15] = 0	
			11T	31	[15] = –0.005	
			10L	10	[10] = –0.005 ; [15] = 0	
			FTN	47	$[11] = FT(591)_A = \alpha$; $[8]_{321} = 592$	
			CL	15	[15] = 0	
			CL	15	[15] = 0	
# 43 →	**# 42**	070	3T	23	$[15] = 10^{-2}\xi_n$	
			10T	30	$[15] = 10^{-2}(\xi_n - 0.5)$	
			N6D6	84		
				07		
				30		
				71	$[6]_{654} = 073$; → 071	
		071	CT	69	→ 073 if [15] ≥ 0 ; [15] = 0	**IF $10^{-2}(\xi_n - 0.5) \geq 0$ → # 45**
	# 43		3T	23	$[15] = 10^{-2}\xi_n$	
			3T	23	$[15] = 2 \times 10^{-2}\xi_n$	
			3L	03	$[3] = 2 \times 10^{-2}\xi_n$; [15] = 0	**$10^{-2}\xi_{n+1} = 2 \times 10^{-2}\xi_n$ → [3]**
			11T	31	$[15] = \alpha$	
			4T	24	$[15] = \alpha + \alpha_n$	
		072	4L	04	$[4] = \alpha + \alpha_n$; [15] = 0	**$\alpha_{n+1} = \alpha_n + \alpha$ → [4]**
			N3D6	83		
				00		
				70	→ 070	**→ # 42**
# 42 →	**# 45**	073	FTN	47	$[11] = FT(592)_A = 10^{-2}$; $[8]_{321} = 593$	
			CL	15	[15] = 0	
			3T	23	$[15] = 10^{-2}\xi_n$	
			11T	31	$[15] = 10^{-2}(1 + \xi_n)$	
			7L	07	$[7] = 10^{-2}(1 + \xi_n)$; [15] = 0	
			3T	23	$[15] = 10^{-2}\xi_n$	
		074	M	41	$[15] = -10^{-2}\xi_n$	
			11T	31	$[15] = 10^{-2}(1 - \xi_n)$	
			÷	63	$[15] = 10^{-2}(1 - \xi_n)/10^{-2}(1 + \xi_n)$	
			3L	03	$[3] = 10^{-2}(1 - \xi_n)/10^{-2}(1 + \xi_n) = x$; [15] = 0	**$x = 10^{-2}(1 - \xi_n)/10^{-2}(1 + \xi_n)$ → [3]**
			11T	31	$[15] = 10^{-2}$	
			SL1	60	$[15] = 10^{-1} = 1/10$	

		075	10L	10	[10] = 1/10 ; [15] = 0	
			3T	23	[15] = x	
			11L	11	[11] = x ; [15] = 0	
			11T	31	[15] = x	
			SR1	32	[15] = x/10	
			×	57	[15] = x^2/10	
		076	7L	07	[7] = x^2/10 ; [15] = 0	
			FTN	47	[11] = $FT(593)_A$ = 1/3 ; $[8]_{321}$ = 594	
			CL	15	[15] = 0	
			7T	27	[15] = x^2/10	
			7T	27	[15] = $2x^2$/10	
			11T	31	[15] = 1/3 + $2x^2$/10	
		077	11L	11	[11] = 1/3 + $2x^2$/10 ; [15] = 0	
			7T	27	[15] = x^2/10	
			×	57	[15] = x^2/10(1/3 + $2x^2$/10)	
			10T	30	[15] = 1/10 + x^2/10(1/3 + $2x^2$/10) = y	**y = 1/10 + x^2/10(1/3 + $2x^2$/10)**
			11L	11	[11] = y ; [15] = 0	
			3T	23	[15] = x	
		078	3T	23	[15] = 2x	
			×	57	[15] = 2xy	
			4T	24	[15] = α_n + 2xy = $10^{-1}\lambda$	**$10^{-1}\lambda = \alpha_n + 2xy$**
			SR3	42	[15] = $10^{-4}\lambda$	
			4L	04	[4] = $10^{-4}\lambda$; [15] = 0	**$10^{-4}\lambda \rightarrow$ [4]**

7.7 Region D

Box 46 has been completed by adding the reference to the line accessed in the numeric function table to the first expression. Otherwise, the code closely matches the Second Run flow diagram.

Region D : Calculate cross-section of material in zone

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 45 →	**# 46**		19T	39	[15] = [19] = i000000ε0k	
		079	S'L1	66	$[12]_1$ = i	
			CL	15	[15] = 0	
			N4D15	73		
				05		
				76	[15] = 0000000576	
			12T	62	$[15]_{321}$ = 576 + i	
		080	8L	08	$[8]_{321}$ = 576 + i = z ; [15] = 0	**z = 576 + i**
			FTN	47	[15] = $FT(z)_B$ = b(z) = 00000b_i0000	
			SR3	42	[15] = 00000000b_i0	
			19T	39	[15] = i000000εb_ik	
			19L	19	[19] = i000000εbk ; [15] = 0	**b = b(z) → $[19]_2$**
			19T	39	[15] = i000000εbk	
		081	S'R2	49	$[12]_{10,9}$ = bk	
			CL	15	[15] = 0	
			12T	62	$[15]_{10,9}$ = bk	
			S'L2	77	$[12]_{21}$ = bk	
			12T	62	$[15]_{21}$ = bk = h	**h = 10b + k**
			N3D8	75		
		082		05		
				00	$[8]_{321}$ = 500	
			8T	28	$[15]_{321}$ = 500 + h	
			8L	08	$[8]_{321}$ = 500 + h	
			FTN	47	[15] = $FT(500 + h)_B$ = 10^{-4}σ(h)	
			10L	10	[10] = 10^{-4}σ(h) ; [15] = 0	**10^{-4}σ(h) → [10]**

7.8 Region E

The code makes a slight optimization in this region, storing the value of α^* in $[2]_{10}$ at the beginning of box 50, instead of repeating the operation in boxes 48 and 49. This code also sets $q^* = [2]_9$ to 0.

There is no reference to box 51 in the code, and the functionality it contains is moved to box 53.

Region E : Determine if terminal event is collision, escape, or census

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 46 →	**# 47**	083	14T	34	[15] = d	
			11L	11	[11] = d ; [15] = 0	
			N6D6	84		
				08		
				70		
				84	$[6]_{654}$ = 087 ; → 084	
		084	10T	30	$[15] = 10^{-4}\sigma$	
			×	57	$[15] = 10^{-4}d\sigma$	
			M	41	$[15] = -10^{-4}d\sigma$	
			4T	24	$[15] = 10^{-4}(\lambda - d\sigma)$	
			CT	69	→ 087 if [15] ≥ 0 ; [15] = 0	**IF $10^{-4}(\lambda - d\sigma) \geq 0$ → # 49**
	# 48		10T	30	$[15] = 10^{-4}\sigma$	
		085	SR1	32	$[15] = 10^{-5}\sigma$	
			7L	07	$[7] = 10^{-5}\sigma$; [15] = 0	
			4T	24	$[15] = 10^{-4}\lambda$	
			SR1	32	$[15] = 10^{-5}\lambda$	
			÷	63	$[15] = 10^{-5}\lambda/10^{-5}\sigma = d_1$	**$d_1 = 10^{-5}\lambda/10^{-5}\sigma$**
			14L	14	$[14] = d_1$; [15] = 0	**d_1 → [14]**
		086	N2D15	72		
				05	$[15]_1$ = 5	
			S'R1	38	$[12]_{10}$ = 5 = α* ; [15] = 0	**5 = α***
			N3D6	83		
				00		
				88	$[6]_{321}$ = 088	**→ # 50**
# 47 →	**# 49**	087	N2D15	72		**($d = d_1$ → [14])**
				04	[15] = 4	
			S'R1	38	$[12]_{10}$ = 4 = α* ; [15] = 0	**4 = α***
			CL	15	[15] = 0	
			CL	15		
			CL	15		
# 48 →	**# 50**	088	2T	22	[15] = abcdefghij	
			SL2	71	[15] = cdefghij00	
			SR2	43	[15] = 00cdefghij	
			12T	62	[15] = α*0cdefghij	
			2L	02	[2] = α*0cdefghij ; [15] = 0	**α* → $[2]_{10}$; 0 = q* → $[2]_9$**
			1T	21	$[15] = t_i + 10^{-3}t_d$ = Miiidddd000	
		089	SL3	70	$[15] = t_d$ = Mdddd000000	
			SR3	42	$[15] = 10^{-3}t_d$ = M999dddd000	
			N6D6	84		
				09		
				90		
				90	$[6]_{321}$ = 099 ; → 090	
		090	11L	11	$[11] = 10^{-3}t_d$; [15] = 0	
			18T	68	[15] = v	
			×	57	$[15] = 10^{-3}t_d v$	
			12L	12	$[12] = 10^{-3}t_d v$; [15] = 0	**$+10^{-3}t_d v$ → [12]**
			14T	34	$[15] = d_1$	
			SR3	42	$[15] = 10^{-3}d_1$	
		091	12T	62	$[15] = 10^{-3}d_1 + 10^{-3}t_d v$	
			CT	69	→ 099 if [15] ≥ 0 ; [15] = 0	**IF $10^{-3}(d_1 + t_d V) \geq 0$ → # 55°**
	# 53		18T	68	[15] = v	
			SR1	32	$[15] = 10^{-1}v$	
			7L	07	$[7] = 10^{-1}v$; [15] = 0	
			14T	34	$[15] = d_1 = d_2$	**($d_2 = d_1$ → [14])**
		092	SR4	53	$[15] = 10^{-4}d_2$	
			÷	63	$[15] = 10^{-4}d_2 / 10^{-1}v = 10^{-3}(d_2/v)$	
			1T	21	$[15] = t_i + 10^{-3}(t_d + d_2/v)$	**$10^{-3}t_d^* = 10^{-3}(t_d + d_2/v)$**
			1L	01	$[1] = t_i + 10^{-3}(t_d + d_2/v)$; [15] = 0	**$10^{-3}t_d^*$ → $[1]_{7654}$**
			CL	15	[15] = 0	
			CL	15	[15] = 0	

# 56' →	**# 54**	093	20T	40	[15] = (r, a)	
			SL5	80	[15] = (a, 0)	
			M	41	[15] = (–a, 0)	
			20T	40	[15] = (r – a, a)	
			7L	07	[7] = (r – a, a) ; [15] = 0	
			14T	34	[15] = d_2	
		094	11L	11	[11] = d_2 ; [15] = 0	
			14T	34	[15] = d_2	
			×	57	[15] = d_2^2	
			13L	13	[13] = d_2^2 ; [15] = 0	
			7T	27	[15] = (r – a, a)	
			7T	27	[15] = 2(r – a, a)	
		095	×	57	[15] = $2(r - a)d_2$	
			13L	13	[13] = $2(r - a)d_2 + d_2^2$; [15] = 0	
			20T	40	[15] = (r, a)	
			11L	11	[11] = (r, a) ; [15] = 0	
			20T	40	[15] = (r, a)	
			×	57	[15] = $r^2 + 2r10^{-5}a + 2(r - a)d_2 + d_2^2$	
		096	√	64	[15] = $\sqrt{(r^2 + 2r10^{-5}a + 2(r - a)d_2 + d_2^2)}$	
			SR4	53	[15] = 0000nnnnnn	
			N2D15	72		
				05	[15] = 0000nnnnn(n+5)	
			SR1	32	[15] = 00000nnnnn' = (0, r*)	**$r^* = \sqrt{(r^2 + 2(r - a)d_2 + d_2^2)}$**
			SL5	80	[15] = (r*, 0)	
		097	20L	20	[20] = (r*, 0) ; [15] = 0	**$r^* \rightarrow [20]_L$**
			7T	27	[15] = (r – a, a) = $r - a + 10^{-5}a$	
			14T	34	[15] = $r - a + 10^{-5}a + d_2$	
			M	41	[15] = $a - r - 10^{-5}a - d_2$	
			20T	40	[15] = $a - r - 10^{-5}a - d_2 + r^*$	
			SR5	52	$[15]_R = a - r - d_2 + r^* = a^*$	**$a^* = a - r - d_2 + r^*$**
		098	20T	40	[15] = (r*, a*)	
			20L	20	[20] = (r*, a*) ; [15] = 0	**$a^* \rightarrow [20]_R$**
			N3D6	83		
				01		
				04	→ 104	**→ # 55**
# 50 →	**# 55°**	099	12T	62	[15] = $10^{-3}t_d v$	
			M	41	[15] = $-10^{-3}t_d v = 10^{-3}d_2$	**$10^{-3}d_2 = -10^{-3}t_d v$**
			N3D6	83		
				01		
				00	→100	
		100	SL3	70	[15] = d_2	
			14L	14	[14] = d_2 ; [15] = 0	**$d_2 \rightarrow [14]$**
			N2D15	72		
				06	$[15]_{21} = 06$	
			S'R1	38	$[12]_{10} = 6$	
			2T	22	[15] = abcdefghij	
		101	SL1	60	[15] = bcdefghij0	
			SR1	32	[15] = 0bcdefghij	
			12T	62	[15] = 6bcdefghij	
			2L	02	[2] = 6bcedfghij ; [15] = 0	**$6 = \alpha^* \rightarrow [2]_{10}$**

	# 56'		1T	21	[15] = “M989180800”	
			S'L3	76	[12] = “M9999999989”	
		102	CL	15	[15] = 0	
			12T	62	[15] = “M9999999989”	
			SL5	80	[15] = “M9998900000”	
			N6D15	74		
				11		
				80		
		103		80	[15] = “M9999018080”	
			SL2	71	[15] = “M9901808000”	**$10^{-3}t_i$* = (t_i + 1)10^{-3} ; .8192 → $10^{-3}t_d$***
			1L	01	[1] = “M9901808000” ; [15] = 0	**$10^{-3}t_d$* → $[1]_{7654}$**
			N3D6	83		
				00		
				93	→ 093	**→ # 54**

7.9 Region F

The starting point of box 56 has been moved slightly for consistency with the code. Otherwise, the Second Run flow diagram and the code are consistent in this region.

Region F : Discriminate between different terminal events

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 54 →	**# 55**	104	N2D15	72		
				06	[15] = 6	
			M	41	[15] = –6	
			4L	04	[4] = –6 ; [15] = 0	
			2T	22	[15] = [2] = α*0cdefghij	
			S'L1	66	$[12]_1$ = α*	
		105	CL	15	[15] = 0	
			12T	62	$[15]_1$ = α*	
			N6D6	84		
				11		
				61		
				06	$[6]_{654}$ = 116 ; → 106	
		106	4T	24	[15] = α* – 6	
			CT	69	→ 116 if [15] ≥ 0 ; [15] = 0	**IF α* – 6 ≥ 0 → # 64**
	# 55*		N6D6	84		
				14		
				61		
				07	$[6]_{654}$ = 146 ; → 107	
		107	N2D15	72		
				05	[15] = 5	
			M	41	[15] = –5	
			12T	62	[15] = α* – 5	
			CT	69	→ 146 if [15] ≥ 0 ; [15] = 0	**IF α* – 5 >= 0 → # 65**
	# 56		N2D15	72		
		108		99	[15] = 0000000099	
			4L	04	[4] = 0000000099	
			19T	39	[15] = i000000εbk	
			S'L1	66	[12] = 000000000i ; [15] = 00000εbk0	
			SR3	42	[15] = 000000000ε	
			12T	62	[15] = 000000000(i + ε)	
		109	4T	24	[15] = i + ε + 99 ; $[15]_1$ = i + ε – 1 = i*	**i + ε – 1 → i***
			S'R1	38	[12] = i*000000000 ; [15] = 00000000x0	
			19T	39	[15] = i000000εb'k	
			SL1	60	[15] = 000000εb'k0	
			SR1	32	[15] = 0000000εb'k	
			12T	62	[15] = i*000000εb'k	
		110	19L	19	[19] = i*000000εb'k ; [15] = 0	**i* → $[19]_{10}$**
	# 57		12T	62	[15] = i*000000000	
			N6D6	84		
				13		
				3 1		
				11	$[6]_{654}$ = 133 or 138 ; → 111	
		111	S'L1	66	[12] = 000000000i* ; [15] = 0	
			12T	62	[15] = 000000000i*	
			M	41	[15] = –i*	
			N2D15	72		
				I	[15] = I – i*	
			CT	69	→ 133 or 138 if [15] >= 0 ; [15] = 0	**IF I – i* >= 0 # 79 or # 82**

7.10 Region M

In box 58, the code copies the original values of i_l and k_l to accumulator 19 as well as that of i_m.

The starting point of box 64 has been changed for consistency with the code.

In box 59, the code does not update the value of γ.

In box 61, the value of s in $[17]_R$ is reset to zero in the course of updating $\overline{n}$. A formula stating this, which appears in the later flow diagram, has been added to the code and flow diagram.

Region M : Print card and restart main loop

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
#64,#73,#80 →	**# 58**	112	GH	55	[11] = [G] ; [15] = [H]	
			CL	15	[15] = 0	
			11T	31	[15] = [G] = "$ii_ik_ii_m$00000k"	
			SL1	60	[15] = "$i_ik_ii_m$00000k0"	
			SR4	53	[15] = "0000$i_ik_ii_m$000"	
			SL3	70	[15] = "0$i_ik_ii_m$000000"	
		113	19T	39	[15] = "$i*i_ik_ii_m$000εbk"	
			19L	19	[19] = "$i*i_ik_ii_m$000εbk" ; [15] = 0	**$i_i*k_i*i_m* = i_ik_ii_m$ → [19]$_{987}$**
# 81 →	**# 58***		N2D15	72		
				06	[15] = 6	
			M	41	[15] = –6	
			4L	04	[4] = –6 ; [15] = 0	
		114	AB	50	[11] = [A] ; [15] = [B]	
			S'L1	66	[12] = 000000000α ; [15] = xxxxxxxxx0	
			N6D6	84		
				12		
				71		
				15	$[6]_{654}$ = 127 ; → 115	
		115	4T	24	[15] = –6	
			12T	62	[15] = α – 6	
			CT	69	→ 127 if [15] ≥ 0 ; [15] = 0	**IF α – 6 ≥ 0 → # 58°**
			N3D6	83		
				01		
				18	→ 118	**→ # 59**
# 55 →	**# 64**	116	N2D15	72		
				02	[15] = 0000000002	
			S'R2	49	[12] = 0200000000 ; [15] = 0	
			12T	62	[15] = 0200000000	
			2T	22	[15] = .2.....	
			2L	02	$[2]_9$ = 2 = q* ; [15] = 0	**q* = 2 → $[2]_9$**
		117	N3D6	83		
				01		
				12	→ 112	**→ # 58**
#58*,#59',#58°→	**# 59**	118	18T	68	[15] = v	
			SR4	53	[15] = 10^{-4}v	
			N2D15	72		
				05	[15] = 10^{-4}v + .5	
			SR1	32	[15] = 10^{-5}v	
			S'R5	58	$[12]_L$ = v ; [15] = 0	
		119	EF	54	[11] = [E] ; [15] = [F]	
			CL	15	[15] = 0	
			11T	31	[15] = [E]	
			SR5	52	$[15]_R = [E]_L = \overline{n}$	
			12T	62	[15] = v + $10^{-5}\overline{n}$	
			18L	18	[18] = v + $10^{-5}\overline{n}$; [15] = 0	**v → $[18]_L$; $\overline{n}$ = n' → $[18]_R$;**
		120	PRINT	45		**Print**
			CL	15	[15] = 0	
			CL	15	[15] = 0	
	# 61		17T	37	[15] = ($\overline{n}$, 0)	
			SR5	52	[15] = (0, $\overline{n}$)	
			N2D15	72		
		121		01	[15] = (0, $\overline{n}$ + 1)	
			SL5	80	[15] = ($\overline{n}$ + 1 , 0)	
			17L	17	[17] = ($\overline{n}$ + 1, 0)	**$\overline{n}$ + 1 → $\overline{n}$; $\overline{n}$ → $[17]_L$; 0 → s → $[17]_R$**

	# 62		N2D15	72		
				02	[15] = P0000000002	
			M	41	[15] = M9999999998	
		122	S'R1	38	[12] = M8000000000	
			CL	15	[15] = 0	
			12T	62	[15] = M8000000000	
			4L	04	[4] = M8000000000 ; [15] = 0	
			2T	22	$[15]_1$ = q	
			S'R1	38	$[12]_{10}$ = q	
		123	CL	15	[15] = 0	
			12T	62	$[15]_{10}$ = q	
			N6D6	84		
				12		
				51		
				24	$[6]_{654}$ = 125 ; → 124	
		124	4T	24	[15] = q – 2	
			CT	69	→ 125 if [15] ≥ 0 ; [15] = 0	**IF q – 2 ≥ 0 → # 63**
			N3D6	83		
				00		
				17	→ 017 or 026	**→ # 2 (I) or # 10 (A)**
# 62 →	**# 63**	125	N2D15	72		
				01	[15] = 1	
			M	41	[15] = –1	
			2T	22	[15] = [2] – 1	
			2L	02	$[2]_{10}$ = $[2]_{10}$ – 1 ; [15] = 0	**q – 1 → q**
			N3D6	83		
		126		00		
				29	→ 029	**→ # 11 (K)**
# 58* →	**# 58°**	127	N2D15	72		
				07	[15] = P0000000007	
			N6D6	84		
				11		
				81		
				28	$[6]_{654}$ = 118 ; → 128	
		128	M	41	[15] = M9999999993	
			12T	62	[15] = α – 7	
			CT	69	→ 118 if [15] ≥ 0 ; [15] = 0	**IF α – 7 ≥ 0 → # 59**
	# 58'		GH	55	[11] = [G] ; [15] = [H]	
			CL	15	[15] = 0	
			11T	31	[15] = [G] = "i.....k"	
		129	S'L1	66	$[12]_1$ = i, [15] = "…….k0"	
			SL4	81	[15] = "....k00000"	
			SL4	81	[15] = "k000000000"	
			SR2	43	[15] = "00k0000000"	
			4L	04	[4] = "00k0000000" ; [15] = 0	
			12T	62	[15] = "000000000i"	
		130	S'R2	49	[12] = "0i00000000"	
			4T	24	[15] = "00k0000000"	
			12T	62	[15] = "0ik0000000"	
			4L	04	[4] = "0ik0000000" ; [15] = 0	
			19T	39	[15] = $i^*i_k i_m 000\varepsilon bk$	
			S'L1	66	$[12]_1$= 000000000i* ; [15]= $i_l k_l i_m 000\varepsilon bk0$	
		131	12T	62	[15] = $i_l k_l i_m 000\varepsilon bki^*$	
			SL2	71	[15] = $i_m 000\varepsilon bki^*00$	
			S'R3	48	[15] = $000i_m 000\varepsilon bk$, [12] = i*000000000	
			12T	62	[15] = $i^*00i_m 000\varepsilon bk$	
			4T	24	[15] = $i^*iki_m 000\varepsilon bk$	
			19L	19	[19] = $i^*iki_m 000\varepsilon bk$	**$i \rightarrow i_l$; $k \rightarrow k_l$**
		132	N3D6	83		
				01		
				18	→ 118	**→ # 59**

7.11 Region N

For the sake of clarity, the layout of this section has been altered in the reconstructed flow diagram.

One apparent error in the code has been corrected in the spreadsheet. At 140.1, on page 17 of the code listing, the logic of the program as described in the flow diagram requires the insertion of a conditional transfer operation to implement the transfer of control defined in the alternative box 82. Elsewhere in the code, alternative boxes are implemented by conditional transfers, so we assume that it has been accidentally omitted at this point. There is space to insert this operation, as row 141 of the function table is not full. On page 17 of the listing, the operation codes in 140 and 141 have been lightly crossed out, suggesting that someone had started to make a correction to the listing at this point but did not carry it through. We have added the missing operation in the spreadsheet, highlighted in red, and adjusted the position of following nine codes in 140 and 141 accordingly, as well as moving the start point of box 83 in the code.

Region N : Zonal escape

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 57, # 84 →	**# 79**	133	N2D15	72		
				31	[15] = 0000000031	
			S'R2	49	[12] = 3100000000 ; [15] = 0	
			2T	22	[15] = α*q*..... = 40.....	
			12T	62	[15] = 71.....	
			2L	02	[2] = 71..... ; [15] = 0	**α* = 7 → $[2]_{10}$; q* = 1 → $[2]_9$**
# 83, # 85 →	**# 80**	134	19T	39	[15] = i*000000εbk	
			SR3	42	[15] = 000i*000000	
			7L	07	[7] = 000i*000000 ; [15] = 0	
			7T	27	[15] = 000i*000000	
			M	41	[15] = –000i*000000	
			5L	05	[5] = –000i*000000 ; [15] = 0	
		135	GH	55	[11] = [G] ; [15] = [H]	
			CL	15	[15] = 0	
			N6D6	84		
				11		
				21		
				36	$[6]_{654}$ = 112 ; → 136	
		136	11T	31	[15] = [G] = $ii_lk_li_m$00000k	
			SL1	60	[15] = $i_lk_li_m$00000k0	
			S'L2	77	[12] = 00000000i_lk_l ; [15] = i_m00000k000	
			SR3	42	[15] = 000i_m00000k	
			5T	25	[15] = i_m – i*	
			CT	69	→ 112 if [15] ≥ 0 ; [15] = 0	**IF i_m – i* ≥ 0 → # 58**
	# 81	137	12T	62	[15] = 00000000i_lk_l	
			S'R3	48	[15] = 0000000000 ; [12] = 0i_lk_l0000000	
			12T	62	[15] = 0i_lk_l0000000	
			N3D6	83		
				01		
				69	→ 169	
		169	7T	27	[15] = 0i_lk_li*000000	
			19T	39	[15] = i*i_lk_li*000εbk	
			19L	19	[19] = i*i_lk_li*000εbk	**i* → i_m → $[19]_7$**
			N3D6	83		
				01		
				13	→ 113	**→ # 58***
# 57 →	**# 82**	138	19T	39	[15] = [19] = i*000000εbk	
			SL1	60	[15] = 000000εbk0	
			SR3	42	[15] = 000000000ε	
			4L	04	[4] = 000000000ε	
			N2D15	72		
				02	[15] = 2	
		139	M	41	[15] = –2	
			4T	24	[15] = ε – 2	
			N6D6	84		
				14		
				21		
				40	$[6]_{654}$ = 142 ; → 140	
		140	CT	69	→ 142 if [15] ≥ 0 ; [15] = 0	**IF ε – 2 ≥ 0 → # 84**
	# 83	141	N2D15	72		
				32	[15] = 0000000032	
			S'R2	49	[12] = 3200000000	
			2T	22	[15] = [2] = α*q*..... = 40.....	
			12T	62	[15] = 72.....	
			2L	02	[2] = 72.....	**α* = 7 → $[2]_{10}$; q* = 2 → $[2]_9$**
			N3D6	83		
				01		
				34	→ 134	**→ # 80**

# 82 →	**# 84**	142	16T	36	[15] = ξ	
			SL3	70	[15] = ξ_7.....000	
			S'L1	66	[12] = 000000000ξ_7	
			CL	15	[15] = 0	
			N2D15	72		
				05	[15] = 5	
		143	M	41	[15] = –5	
			12T	62	[15] = ξ_7 – 5	
			N6D6	84		
				13		
				31		
				44	$[6]_{654}$ = 133 ; → 144	
		144	CT	69	→ 133 if [15] ≥ 0 ; [15] = 0	**IF ξ_7 – 5 ≥ 0 → # 79**
	# 85		N2D15	72		
				02	[15] = 2	
			S'R1	38	$[12]_{10}$ = 2	
			2T	22	[15] = [2] = ...	
			SL1	60	[15] = ...0	
		145	SR1	32	[15] = 0.....	
			12T	62	[15] = 2.....	
			2L	02	[2] = 2..... ; [15] = 0	**α* = 2 ; α* → $[2]_{10}$**
			N3D6	83		
				01		
				34	→ 134	**→ # 80**

7.13 Region H

The beginning of box 65 sets the return address for the subroutine call that precedes it. As previously, this is shown in the reconstructed flow diagram in an unnumbered box. The subroutine returns to row 197 of the function table before returning to the regular program sequence at line 147: it looks as if this was done to insert the order to clear accumulator 15 at 197.3 at a late stage in the development of the program.

A slight amendment has been made to box 65 to emphasize that the initial value of g is stored in [10].

In box 66, digits of ξ are not in fact stored in [3].

In box 68, the code does not in fact store the value of g_2 in [10]. This would cause a problem if the special process was included, as the value of g_3 would then be incorrectly calculated. We have therefore assumed that this is an error in the code, and left the original formula in box 68.

The three unnumbered boxes below box 68 in the reconstructed flow diagram describe the special process code, and are adapted from the corresponding boxes in the later flow diagram. Note that g_3 is stored in [10], corroborating the suspicion that a coding error was made in box 68.

Monte Carlo Second Run
Region H : Determine collision type

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 55* →	**# 65**	146	D	99	delay	
			D	99	delay	
			N6D6	84		
				19		
				71		
				71	$[6]_{654} = 197 = \omega_2$; → 171	**ω_2 → ω ; → ρ**
ω_2 →		197	19T	39	[15] = i000000εbk	
			S'R2	49	[12] = bk00000000	
			CL	15	[15] = 0	
			N3D6	83		
				01		
				47	→ 147	
		147	12T	62	[15] = bk00000000	
			SR4	53	[15] = 0000bk0000	
			SR4	53	[15] = 00000000bk	
			N4D15	73		
				05		
				00	[15] = 00000005bk	
		148	8L	08	$[8]_{321}$ = 5bk ; [15] = 0	**h = 5bk → [8]**
			16T	36	[15] = ξ	
			SL4	81	[15] = "6543210000"	
			S'L3	76	$[12]_{321}$ = 654, [15] = "3210000000"	
			4L	04	[4] = "3210000000" ; [15] = 0	**ξ_{321} → $[4]_{10,9,8}$**
			12T	62	[15] = "0000000654"	
		149	M	41	[15] = –654 = g	
			10L	10	[10] = g ; [15] = 0	**g = $-\xi_{654}$ → [10]**
	# 66		FTN	47	$[11] = FT(5bk)_A$; $[8]_{321}$ = 5bk + 1	
			CL	15	[15] = 0	
			11T	31	$[15] = FT(5bk)_A$ = "1461370000"	
			S'L3	76	$[15]_{321}$ = "137", $[12]_{321}$ = "146"	
		150	5L	05	[5] = "1370000000" ; [15] = 0	
			12T	62	[15] = "0000000146" = σ_E/σ_t	
			N6D6	84		
				15		
				81		
				51	$[6]_{654}$ = 158 ; → 151	
		151	10T	30	$[15] = \sigma_E/\sigma_t + g = g_1$	**$g_1 = g + \sigma_E/\sigma_t$**
			10L	10	$[10] = g_1$; [15] = 0	**g_1 → [10]**
	# 67		10T	30	$[15] = g_1$	
			CT	69	→ 158 if [15] ≥ 0 ; [15] = 0	**IF $g_1 \geq 0$ → # 74**
	# 68		5T	25	[15] = "1370000000"	
			SR5	52	[15] = "0000013700"	
		152	SR2	43	[15] = "0000000137" = σ_I/σ_t	
			10T	30	$[15] = g_1 + \sigma I/\sigma_t = g_2$	**$g_2 = g_1 + \sigma_I/\sigma_t$ → [10]**
	# 69		N6D6	84		
				16		
				41		
				53	$[6]_{654}$ = 164 ; → 153	
		153	CT	69	→ 164 if [15] ≥ 0 ; [15] = 0	**IF $g_2 \geq 0$ → # 77**
			N3D6	83		
				01		
				54	→ 154 (190 for S.P.)	**→ # 70 or S.P.**

	S.P.	190	19T	39	[15] = i000000ϵbk	
			S'L1	66	$[12]_1$ = i ;, [15] = 000000ϵbk0	
			N6D6	84		
				15		
				41		
				91	$[6]_{321}$ = 154 ; → 191	
		191	CL	15	[15] = 0	
			N2D15	72		
				02	[15] = 2	
			M	41	[15] = –2	
			12T	62	[15] = i – 2	
			CT	69	→ 154 if [15] ≥ 0 ; [15] = 0	**IF i – 2 ≥ 0 → # 70**
		192	N2D15	72		
				86	[15] = 86	
			8T	28	[15] = 5bk + 1 + 86	
			8L	08	$[8]_{321}$ = 587 + bk ; [15] = 0	
			FTN	47	$[15] = FT(587 + bk)_R = \sigma_{2H}/\sigma_t$	
			10T	30	$[15] = \sigma_{2H}/\sigma_t + g_1 = g_3$	
		193	10L	10	$[10] = g_3$	**$g_3 = g_2 + \sigma_{2H}/\sigma_t$ → [10]**
			10T	30	$[15] = g_3$	
			N6D6	84		
				17		
				81		
				94	$[6]_{321}$ = 178 ; → 194	
		194	CT	69	→ 178 if [15] ≥ 0 ; [15] = 0	**IF g_3 ≥ 0 → # 52***
			N3D6	83		
				01		
				54	→ 154	**→ # 70**

7.14 Region L

In the reconstructed flow diagram, box 73 has been split into two boxes, reflecting a coding optimization that meant that control actually passed from box 72 to the middle of box 73. The layout of this section has been altered for clarity.

Region L : Absorption and fission

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 69 →	**# 70**	154	N2D15	72		
				45	[15] = 45	
			8T	28	[15] = 5bk + 1 + 45	
			8L	08	$[8]_{321}$ = 5bk + 46 = p ; [15] = 0	**p = 46 + bk**
			FTN	47	[11] = FT(546 + bk)$_A$ = v(p)	
			CL	15	[15] = 0	
		155	11T	31	[15] = v	**v = v(p) → [15]**
	# 71		S'L1	66	$[12]_1 = v_{Int}$; $[15] = v_{Dec}$	**v_{Int}(p) → q* → $[12]_1$; v_{Dec} → [15]**
	# 72		N6D6	84		
				15		
				71		
				56	$[6]_{654}$ = 157 ; → 156	
		156	M	41	$[15] = -v_{Dec}$	
			4T	24	$[15] = \xi_{321} - v_{Dec}$	
			CT	69	→ 157 if [15] ≥ 0 ; [15] = 0	**IF $\xi_{321} - v_{Dec} \geq 0$ → 157**
	# 73		CL	15	[15] = 0	
			N2D15	72		
				01	[15] = 1	**q* + 1 → q***
# 72 →		157	12T	62	$[15]_1$ = q* +? 1	
			S'R2	49	$[12]_9$ = q* +? 1 ; [15] = 0	
			12T	62	$[15]_9$ = q* +? 1	
			N3D6	83		
				01		
				70	→ 170	
		170	2T	22	[15] = [15] + [2] = α*q*00000000	
			2L	02	[2] = α*q*00000000 ; [15] = 0	**q* → $[2]_9$**
			N3D6	83		
				01		
				12	→ 112	**→ # 58**

7.15 Region JK

In box 75, the two formulas from the flow diagram have been merged into one, as the intermediate variable does not appear in the code.

In box 76, the code also reinitializes the values of ε and b. The formulas describing this have been moved from the specification box immediately following.

The first line of 77 in the flow diagram is rather unclear. The operation performed by the code is described by a simpler formula, taken from the later flow diagram.

Box 52* is part of the special process and does not appear in the Second Run flow diagram. The calculations performed are very similar to those described in box 52* of the First Run flow diagram, and the formulas in the reconstructed flow diagram are derived from those.

There appears to be a coding error at this point: if the special process is invoked, the value of a^{**} calculated in box 52* will be overwritten in box 75. The code gives no indication of how this might have been avoided.

Region JK : Elastic and Inelastic Scattering

Label	FT	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
# 67 →	**# 74**	158	N2D15	72		
				66	[15] = 0000000066	**$\theta_1 \rightarrow \theta$**
			S'R4	59	[12] = 0066000000 ; [15] = 0	
			12T	62	[15] = 0066000000	
			8L	08	[8] = 0066000000 ; [15] = 0	**$\theta \rightarrow [8]_{987}$**
			CL	15	[15] = 0	
# 78 →	**# 75**	159	20T	40	[15] = (r*, a*)	
			SR5	52	[15] = (0, r*)	
			SL5	80	[15] = (r*, 0)	
			20L	20	[20] = (r*, 0) ; [15] = 0	
			20T	40	[15] = (r*, 0)	
			20T	40	[15] = (2r*, 0)	
		160	11L	11	[11] = (2r*, 0) ; [15] = 0	
			4T	24	[15] = ξ_{321}	
			×	57	[15] = $2r^*\xi_{321}$	
			SR5	52	$[15]_R = 2r^*\xi_{321}$	
			20T	40	[15] = (r*, $2r^*\xi_{321}$)	
			20L	20	[20] = (r*, $2r^*\xi_{321}$) ; [15] = 0	**$a^{**} = 2r^*\xi_{321} \rightarrow [20]_R$**
	# 76	161	N2D15	72		
				01	[15] = 1	
			17T	37	[15] = (n, s) + 1	
			17L	17	[17] = (n, s + 1) ; [15= 0	**$s + 1 \rightarrow s \rightarrow [17]_{321}$**
			19T	39	[15] = i000000εbk	
			S'R5	58	[15] = 00000i0000 , [12] = 00εbk00000	
		162	12T	62	[15] = 00εbki0000	
			SL4	81	[15] = ki00000000	
			S'L1	66	$[12]_1$ = k , [15] = i000000000	
			12T	62	[15] = i00000000k	
			19L	19	[19] = i00000000k ; [15] = 0	**$0 \rightarrow b \rightarrow [19]_2$, $0 \rightarrow \varepsilon \rightarrow [19]_3$**
			N3D6	83		
		163		00		
				42	→ 042	**→ # 21**
# 69 →	**# 77**	164	4T	24	[15] = $\xi_3\xi_2\xi_1$0000000	
			SL2	71	[15] = $\xi_1$000000000	
			SR4	53	[15] = 0000$\xi_1$00000	
			SR4	53	[15] = 00000000$\xi_1$0	
			DS	46	[15] = 00000000$\xi_1$0	
			N2D15	72		
		165		05	[15] = 00000000$\xi_1$5 = $10^{-8}\phi$	**$\phi = \xi_1 + 0.5$**
			SL5	80	[15] = $10^{-3}\phi$	
			11L	11	[11] = $10^{-3}\phi$; [15] = 0	
			11T	31	[15] = $10^{-3}\phi$	
			11T	31	[15] = $2 \times 10^{-3}\phi$	
			11T	31	[15] = $3 \times 10^{-3}\phi$	
		166	11T	31	[15] = $4 \times 10^{-3}\phi$	
			11T	31	[15] = $5 \times 10^{-3}\phi$	
			√	64	[15] = $\sqrt{(5 \times 10^{-3}\phi)}$	
			SL1	60	[15] = $10 \times \sqrt{(5 \times 10^{-3}\phi)} = \sqrt{(½\phi)}$	
			11L	11	[11] = $\sqrt{(½\phi)}$; [15] = 0	
			18T	68	[15] = v	
52* →		167	×	57	[15] = $v\sqrt{(½\phi)} = v^*$	**$v^* = v\sqrt{(½\phi)}$**
			18L	18	[18] = v* ; [15] = 0	**$v^* \rightarrow [18]$**
	# 78		N2D15	72		
				56	[15] = 0000000056	**$\theta_2 \rightarrow \theta$**
			S'R4	59	[12] = 00560000000 ; [15] = 0	
			12T	62	[15] = 00560000000	
		168	8L	08	$[8]_{987}$ = 056	**$\theta \rightarrow [8]_{987}$**
			N3D6	83		
				01		
				59	→ 159	**→ # 75**

S.P. →	**# 52***	178	4T	24	$[15] = \xi_{321}0^7$	
			S'L1	66	$[12] = 0^9\xi_3$; $[15] = \xi_2\xi_1 0^8 = \delta$	
			10L	10	$[10] = \delta$; $[15] = 0$	**$\delta = \xi_2\xi_1$ → [10]**
			12T	62	$[15] = 0^9\xi_3$	
			4L	04	$[4] = 0^9\xi_3$; $[15] = 0$	
			CL	15	$[15] = 0$	
		179	CL	15	$[15] = 0$	
			20T	40	$[15] = (r^*, a^*)$	
			S'L5	86	$[12]_R = r^*$, $[15] = a^*$	
			11L	11	$[11] = a^*$; $[15] = 0$	
			12T	62	$[15]_R = r^*$	
			SL5	80	$[15] = (r^*, 0)$	
		180	20L	20	$[20] = (r^*, 0)$; $[15] = 0$	
			11T	31	$[15] = a^*$	
			M	41	$[15] = -a^*$	
			20T	40	$[15] = r^* - a^*$	
			3L	03	$[3] = r^* - a^*$; $[15] = 0$	
			3T	23	$[15] = r^* - a^*$; $[15] = 0$	
		181	×	57	$[15] = a^*(r^* - a^*)$	
			13L	13	$[13] = a^*(r^* - a^*)$	
			20T	40	$[15] = r^*$	
			×	57	$[15] = a^*r^* + a^*(r^* - a^*)$	
			11L	11	$[11] = a^*r^* + a^*(r^* - a^*)$ $[15] = 0$	
			10T	30	$[15] = \delta$	
		182	×	57	$[15] = \delta(a^*r^* + a^*(r^* - a^*))$	
			11L	11	$[11] = \delta(a^*r^* + a^*(r^* - a^*))$; $[15] = 0$	
			10T	30	$[15] = \delta$	
			M	41	$[15] = -\delta$	
			DS	46	$[15] = 1 - \delta$	
			×	57	$[15] = \delta(1 - \delta)(a^*r^* + a^*(^*r - a^*))$	
		183	√	64	$[15] = \sqrt{}(\delta(1 - \delta)(a^*r^* + a^*(r^* - a^*)))$	
			14L	14	$[14]=\sqrt{}(\delta(1 - \delta)(a^*r^* + a^*(r^* - a^*)));[15]=0$	
			10T	30	$[15] = \delta$	
			√	64	$[15] = \sqrt{}\delta$	
			SR2	43	$[15] = 10^{-2}\sqrt{}\delta$	
			7L	07	$[7] = 10^{-2}\sqrt{}\delta$; $[15] = 0$	
		184	N4D15	73		
				05		
				56	$[15] = 0^7556$	
			4T	24	$[15] = 0^7556 + 0^9\xi_3$	
			8L	08	$[8]_{321} = 556 + \xi_3$	
			FTN	47	$[15] = c(\xi_3) = \gamma$	
		185	11L	11	$[11] = \gamma$; $[15] = 0$	**$\gamma = c(\xi_3)$ → [11]**
			14T	34	$[15] = \sqrt{}(\delta(1 - \delta)(a^*r^* + a^*(r^* - a^*)))$	
			×	57	$[15] = \gamma\sqrt{}(\delta(1 - \delta)(a^*r^* + a^*(r^* - a^*)))$	
			13L	13	$[13]=\gamma\sqrt{}(\delta(1-\delta)(a^*r^* + a^*(r^* - a^*)));[15]=0$	
			10T	30	$[15] = \delta$	
			11L	11	$[11] = \delta$; $[15] =0$	
		186	3T	23	$[15] = r^* - a^*$	
			×	57	$[15]=\delta(r^*-a^*)+\gamma\sqrt{}(\delta(1-\delta)(a^*r^*+a^*(r^*-a^*)))$	**$\Upsilon = \delta(r^*-a^*) + \gamma\sqrt{}(\delta(1-\delta)(a^*r^* + a^*(r^*-a^*)))$**
			SR2	43	$[15] = 10^{-2}\Upsilon$	
			÷	63	$[15] = 10^{-2}\Upsilon / 10^{-2}\sqrt{}\delta$	
			M	41	$[15] = -10^{-2}\Upsilon / 10^{-2}\sqrt{}\delta$	
			20T	40	$[15] = r^* - 10^{-2}\Upsilon / 10^{-2}\sqrt{}\delta$	
		187	SR5	52	$[15]_R = r^* - 10^{-2}\Upsilon / 10^{-2}\sqrt{}\delta = a^{**}$	
			20T	40	$[15] = (r^*, a^{**})$	
			20L	20	$[20] = (r^*, a^{**})$; $[15] = 0$	**$a^{**} = r^* - 10^{-2}\Upsilon / 10^{-2}\sqrt{}\delta$ → $[20]_R$**
			7T	27	$[15] = 10^{-2}\sqrt{}\delta$	
			SL2	71	$[15] = \sqrt{}\delta$	
			11L	11	$[11] = \sqrt{}\delta$; $[15] = 0$	
		188	18T	68	$[15] = \sqrt{}$	
			N3D6	83		
				01		
				67	→ 167	**→ 167**
			(×)	*(57)*	*([15] = v√δ = v*)*	
			(18L)	*(18)*	*([18] = v* ; [15] = 0)*	***(v* = v√δ → [18])***

7.16 Region G

The code listing suggests that significant changes were made to the code in this region in the course of development, particularly in row 172, where a number of instructions have been replaced by "dummy" codes. It is not possible to see what the original codes were, however.

Region G : Refresh random number

Label	Box	FT	Symbol	Code	Effect on accumulator contents	Formulas from flowchart boxes
(ρ) 66, 146 →		171	CL	15	[15] = 0	
			16T	36	[15] = abcdefghiji = ξ'	
			S'R1	38	[15] = 0abcdefghi ; [12] = j000000000	
			12T	62	[15] = jabcdefghi = ξ	**$\xi = \Theta^{-}\xi'$**
			S'L5	86	[12]=0^5jabcd=$10^{-5}\xi_0$; [15]=efghi0^5=$10^5\xi_1$	**$\xi_0 = \xi_{10-6}$; $\xi_1 = \xi_{5-1}$**
			11L	11	[11] = $10^5\xi_1$ **; [15] = 0**	
		172	12T	62	[15] = $10^{-5}\xi_0$	
			10L	10	[10] = $10^{-5}\xi_0$	
			CL	15	[15] = 0	
			CL	15	[15] = 0	
			CL	15	[15] = 0	
			CL	15	[15] = 0	
		173	12T	62	[15] = $10^{-5}\xi_0$	
			SL5	80	[15] = ξ_0	
			×	57	[15] = $\xi_0(10^5\xi_1) = \overline{\xi}_0\overline{\xi}_1$	**$\overline{\xi}_0\overline{\xi}_1 = \xi_0(10^5\xi_1)$**
			7L	07	[7] = $\overline{\xi}_0\overline{\xi}_1$	
			11T	31	[15] = $10^5\xi_1$	
			×	57	[15] = $(10^5\xi_1)^2$	
		174	SR5	52	[15] = $10^{-5}\overline{\xi}_1^{\,2}$	**$\overline{\xi}_1^{\,2}$ = DIG (10–6) of $(10^5\xi_1)^2$**
			7T	27	[15] = $\overline{\xi}_0\overline{\xi}_1 + 10^{-5}\overline{\xi}_1^{\,2}$	
			7T	27	[15] = $2\overline{\xi}_0\overline{\xi}_1 + 10^{-5}\overline{\xi}_1^{\,2}$	
			7L	07	[7] = $2\overline{\xi}_0\overline{\xi}_1 + 10^{-5}\overline{\xi}_1^{\,2}$	
			10T	30	[15] = $10^{-5}\xi_0$	
			SL5	80	[15] = ξ_0	
		175	11L	11	[11] = ξ_0	
			11T	31	[15] = ξ_0	
			×	57	[15] = $\xi_0^{\,2}$	
			SL5	80	[15] = $10^5\overline{\xi}_0^{\,2}$	**$\overline{\xi}_0^{\,2}$ = DIG (5–1) of $\xi_0^{\,2}$**
			7T	27	[15] = $10^5\overline{\xi}_0^{\,2} + 2\overline{\xi}_0\overline{\xi}_1 + 10^{-5}\overline{\xi}_1^{\,2} = \xi'$	**$\xi' = 10^5\overline{\xi}_0^{\,2} + 2\overline{\xi}_0\overline{\xi}_1 + 10^{-5}\overline{\xi}_1^{\,2}$**
			DS	46	[15] = ξ' = jabcdefghi	
		176	S'L1	66	[12] = 000000000j, [15] = abcdefghi0	
			12T	62	[15] = abcdefghij = ξ	
			16L	16	[16] = ξ	**$\xi = \Theta\xi'$ → [16]**
			COUNT	***N***	Halt after 3000 squarings	Special order: cf AR:22
			CT	69	→ ω	→ ω

8 Observations and conclusions

The code given in the listing seems to be a complete and plausible program corresponding very closely to the Second Run diagram. We suspect but cannot prove that the code listing used here is the specific copy of the code used during the execution of the Second Run. Presumably Metropolis and von Neumann would have wanted to keep a master copy of the code as debugged and executed, and we see evidence of a careful process of erasure, addition or correction in many parts of the document. For example, the corrections in the listing and the insertions at rows 195 – 197. On the other hand, the presence of at least one error in the code (discussed earlier) may indicate that this is not the final copy used during execution.

The basic functioning and correctness of the code could be further tested by running the code on the ENIAC emulator. We have successfully emulated the set-up that implemented the code used in mid-1948, so in principle this would be possible. Some additional capabilities would be needed to simulate the manual steps involved in sorting punched cards. The main stumbling block is that we don't have the numerical data that would let us produce meaningful results or be sure that the program was functioning properly. We know the mathematical significance of the constants and parameters set on FT3 but not the actual content of each location. One or two actual values are scattered around the *Actual Running* report and in the annotations on the code listing, but not enough to fill up FT3 with realistic data. It seems from the archival sources that the classified nuclear cross section data needed to set up the problem was protected far more carefully than the program code or flowcharts so this is almost certainly deliberate.[8]

[8] Discussed on page 195 of *ENIAC In Action*.