"Actual Running of the Monte Carlo Problems on the ENIAC"

Introduction to the 2014 Reconstruction

This document reconstructs a 1949 report produced collaboratively by Klara von Neumann, John von Neumann, and Nick Metropolis describing the two Monte Carlo runs made in in March-April and October-November 1949. The history of these calculations is described in Thomas Haigh, Mark Priestley & Crispin Rope, "Los Alamos Bets on ENIAC: Nuclear Monte Carlo Calculations, 1947-8," *IEEE Annals of the History of Computing* 36:1 (Apr-Jun 2014) which is available with other related materials from www.EniacInAction.com.

The final typewritten version of the report was among the evidence collected for the ENIAC lawsuit Honeywell v. Sperry Rand tried in 1972. Specifically this was Plaintiff's Trial Exhibit No. 22798. Unfortunately only a copy of the front page was held in the main file subsequently archived at the University of Pennsylvania and microfilmed for access at other locations. This copy included a note "A complete copy of this document is located in our plantiff's file in the court room."[1] The front page did not specify authors, dates, or other identifying information.

Fortunately several draft versions are preserved in Papers of John von Neumann, Manuscripts Division, Library of Congress, Washington, box 12, folder 6. The earliest version, "III: Actual Technique – The Use of the ENIAC" is a 17 page manuscript handwritten by Klara von Neumann. It covered only the "first run" completed in April 1948 and so was likely written in mid-1948. We are not sure what "III" meant – perhaps it was intended as the third part of a bundle of technical documentation for submission to Los Alamos. The same folder holds a typewritten version with insertions and corrections by John von Neumann numbered x1 to x83 noted in the right margin. Eight larger passages of handwritten text on separate sheets are marked for insertion at various points.

The collaborators later returned to "Actual Technique" to create an updated version called "Actual Running of the Monte Carlo Problems on the ENIAC" that also covers the "Second Run" version of the program. The same folder holds three full versions of "Actual Running…" one in Klara von Neumann's hand and the other two typewritten. One 22 page typescript had symbols added by hand annotated to all pages and other corrections noted, primarily by Klara. It was followed by three pages of handwritten further corrections, with page and line numbers noted for each. We based this reconstruction on that fully corrected typescript, incorporating all the corrections indicated the notes. The pagination of the typescript has been preserved as has the original formatting. Our aim was to produce a usable approximation of the lost final version rather than to chart the evolution of its content, so no attempt has been made to differentiate between the original typescript and the handwritten corrections.

We date completion of this report to 1949 for two reasons. First, it mentions that a third run, not described, had taken place. We know that to have taken place in 1949. Second, Metropolis wrote to Klara von Neumann in September 1949, "Here is your manuscript together with a rough typewritten copy… The flow diagrams will definitely be finished on Monday and will be sent to you on that day."[2]

This reconstruction was created by Ann Graf, who did most of the transcription work, and by Mark Priestley who checked it and added the mathematical symbols. Thanks to Marina von Neumann Whitman who gave permission to reproduce this and other materials from her father's papers for our project.

Thomas Haigh, June 2014.

---

[1] It is possible that a complete copy exists in Metropolis' personal files at Los Alamos, but like all Los Alamos records these are now off limits to outside researchers except via Freedom of Information Act requests. I filed several such requests for specific documents two years ago and am still waiting.

[2] Letter, Metropolis to K. von Neumann, September 23, 1949, box 19, folder 7 of the Library of Congrees collection of von Neumann's papers.

ACTUAL RUNNING OF THE MONTE CARLO PROBLEMS ON THE ENIAC

Up to the present date, three Monte Carlo runs have been made on the ENIAC,[3] i.e., three separate groups of problems have been computed, all based on the Monte Carlo method, but with some changes between runs in the logical sequence (flow-diagram) and the actual computations (operation boxes [4] within the flow-diagram). The changes were made partly because of experience gained during the previous run for more efficient use of the machine on this type of problems, partly because of the somewhat different requirements necessitated by the group of problems under consideration.

We shall attempt here to describe the details of the flow-diagram and operation boxes of the First run. We will then point out the divergences and the reasons for these changes in the Second Monte Carlo run. (The Third Monte Carlo run will be described in a separate report.) In order to start the computation, an IBM card, representing one neutron, was read into the constant transmitter. In the First run each card contained the following data:

α indicating the event that terminates the period of life (of the neutron) to which the card refers.

$\alpha =$ 3 = Absorption)
4 = Total Escape)
5 = Fission)
6 = Census)

3 Similar Monte Carlo computations have been made on the ENIAC for the Argonne Laboratories by Dr. M. Mayer and her group.

[4] Actual operation boxes are on the Flow diagram identified by numerals; boxes identified by some other symbols are either "explanation boxes" or listings of quantities as stored in any one of the twenty accumulators at a particular point in the computation.

ξ the ten digit random number last used.

i the zone in the spherical assembly where the event occurred. (In the First run zone limits were chosen according to the changing material composition in the assembly. In some of the problems of the Second run, bands of the same material composition were split up into several zones in order to facilitate the calculations. (An explanation of this method will be given later in this report.)

t the time in units of 10^{-8} seconds (1 shake) when the event occurred.

V the velocity of the neutron in m/shake at the time, t.

r the distance in units of meters from the center of the sphere at the time,

$a = r(1 - \cos\theta)$ a convenient parameter defining the direction of the neutron's path; θ is the polar angle.

q indicating the number of neutrons that originated in the terminal event and that are initiated by the card now under consideration; it is essentially its weight, i.e. the instruction to the machine how many times to use the data from this card and follow its course to the terminal event, before reading the data from the next card in the stack. $q = 1$ for each event, except if the collision was fission, when it may be any integer $\geqq 1, \leqq 4$, according to the number of neutrons created by fission – this number being determined by a random process, as provided by the Monte Carlo method.

In addition, α^* and q^* were also noted on the card, which was printed after having determined the terminal event for the period of life of the neutron under consideration, and calculations completed for the new values of the characteristic quantities (V, r, a and t).

α^* indicating the event that terminates the period of life (of the neutron) to which the present calculation and the card to be printed refer. This will become α when the card now printed will be read.

q^* indicating the number of neutrons that will originate in the terminal event of the period of life (of the neutron) to which the present calculation and the card to be printed refer. This will become q when the card now printed will be read.

Apart from the above listed quantities that are necessary for the logical sequence and for the numerical computation, we also indicated four "family" or identification numbers:

g the fission generation.

n^+ the "parent" neutron, i.e. the card from which the present card originated.

n^* the original "ancestor" card from which the present card originated.

$\bar{n}$ a serial number which is increased by one after each printing. This quantity becomes n', the "parent" neutron at the beginning of the calculation which determines the next terminal event in the succeeding period of life of the neutron family.

These identification numbers were not needed for the actual computation, but were helpful to trace the genealogy of any neutron in the sample population.

In the Second run it seemed to be useful to carry some further "informative" quantities on the card:

i_l the zone where the previous census (of the neutron) occurred.

i_m the zone farthest away from the center which any member of the genealogy ever reached during its total life-span.

k^* the velocity interval in which the neutron travelled at the time of the terminal event.

k_l the velocity interval of the neutron at the previous census.

s the number of scatterings before reading one of the terminal events.

These quantities were helpful in tracing the history of a neutron or neutron family, in particular, they gave some indication of the "flux" in the sample population of the assembly under consideration.

To start any of the problems of the First run, one hundred cards were read, each representing one neutron originated by fission, with a random velocity distributed over the fission spectrum, represented by the "centers of gravity" $V(\xi)$, $(\xi = 0, 1, \ldots, 9)$, of an appropriate subdivision of that spectrum into suitable velocity intervals. All initial neutrons were started at the center of

the spherical assembly at zero time. These neutrons then, in turn, produced other neutron cards, indicating the event that terminated the calculation of their path, one of the events being, that a census time T, has been reached. Total escape and absorption cards were then sorted out and removed from the newly printed stack since their path did not have to be followed any longer. Fission cards, representing neutrons which produced two or three new neutrons, (n.b. in the problems here described only two or three way fission was considered) and whose time, t was still with census time, were then put back into the reader until each newly printed card was either a total escape, absorption or census card, the last indicating that all neutrons that survived have reached the end of the census time interval. At this point, since there were no more cards to be read, the automatic iterative process of the machine was interrupted, the machine stopped and a new cycle was started with a new value for T, the next census time. To keep an exact one hundred card sample for each census interval, an appropriate number of the census cards were either duplicated by a random choice, if the stack had decreased, which happened normally (but of course not necessarily or exclusively) when the assembly was under-critical, or discarded if the stack had increased, which was similarly related to the assembly being critical or super-critical. These cycles, i.e. census time intervals, could be continued as many times as desired, using each time the census cards, obtained from the end of the previous cycle, as input data. The method described above was used, however, only in the First Monte Carlo run. An attempt was made in the Second Monte Carlo run, to include in the logical sequence and coding of the program, an automatic way of handling the beginning and

ending of a time cycle, i.e. the individual neutrons belonging to different time cycles could be calculated instead of having to wait until all surviving neutrons have been driven to the end of the same census interval. With this method no attempt was made to keep the input stack at a fixed number, however, in order to insure a sufficiently great sample population at the end of the calculation, all neutrons surviving the end of a census interval, were given a double weight before printing, (i.e. $q^* = 2$), thereby doubling the surviving neutron population at the beginning of the next time cycle. The original input cards were also treated differently in the Second Monte Carlo run, further distinguishing between the under-critical and critical or near-critical assemblies. An attempt was made to make more efficient use of the machine computing time, i.e., to increase the sample population as the neutrons in the assembly approach, with time, the expected normal spatial and energy distribution. In the under-critical cases, assuming a neutron source at the center, the original or "virgin" input fission neutrons were again started at the center of the assembly, with a random fission velocity but spaced in time equal intervals, γ. At each census, T, the value of γ was halved, thereby doubling the number of "virgin" cards to be read in the next time cycle. This had the intended effect that half of all the "virgin" neutrons thus generated were read and their path computed in the last or latest census-cycle. (See Monte-Carlo Flow Diagram II; boxes 2 ... 7.)

In the case of the critical or super-critical assemblies, a method similar to the First run was used, starting, however, with a small initial input stack of about twenty neutrons at same original T, but effecting the increase of the sample population, (besides its natural increase), by doubling at the end of each census cycle as described above.

The Flow-Diagrams

For all six problems, in the First Monte Carlo run, one common Flow-Diagram and one common code, i.e., logical sequence was used. Accordingly, only the one Function Table containing the numerical data specific to the assembly to be computed, had to be changed from one problem to the other. T, the census time interval was taken as 1 shake $= 10^{-8}$ sec for all problems in this run; the value of T was set by an external switch on the Constant Transmitter and manually increased by one at the end of each time-cycle.

A card, representing a neutron, was first examined for α, i.e. the type of terminal event which occurred at its previous period of life. If the event was Absorption, ($\alpha = 3$) or Total Escape, ($\alpha = 4$) the machine discarded the card and read the next one in the stack. If it was a census card, ($\alpha = 6$) belonging to the time-cycle presently being computed, r, a and V were placed into the appropriate accumulators, assigned for temporary storage for these particular quantities; i.e., census cards were started at the same point in the assembly with the same direction and velocity at which they terminated at the end of the previous census period.

If the card represented a fission neutron ($\alpha = 5$), the value of r was read from the card, but a and V were chosen by certain random processes; using the given r and a random number equidistributed between 0 and 1 and picking out, with the help of a similarly equidistributed random number, one of the ten $V(\xi)$, the "center of gravity" velocity lying between certain intervals of the fission spectrum range of velocity.

The ten values for $V(\xi)$ were set on the numerical Function Table at the beginning of the run. (For explanation of the method applied to obtain the random digits used throughout the problem, see pages $21 - 22$.) The values a and V that had been thus computed, were then stored as in the census case. After this the fission and census neutron's path merge and t and i are stored in their assigned accumulator locations. (For above described section of the Flow-Diagram I, see operation boxes: 1.3*, 1*, 1.2*, 2*, 2.1*, 3*, 5*, 7* and 8*).

The Second Monte Carlo run, (Flow Diagram II), except for a few simplifications, does not differ up to this point essentially from the logical set-up and computations as described above for Flow diagram I: (see: operation box 11, 12, 14 and 16.) Operation box 10, replaces the sorting on α for those cards which are representing neutrons whose paths do not have to be followed any longer, (the terminal event in their previous period of life having been Absorption, Total Escape or, as we shall explain later, in some cases, Zonal Escape). In these instances $-q = 0$ and, since the ENIAC interprets this as a positive integer, the card was automatically discarded and a new card immediately read from the stack. Also, since in this run we did not try to separate census cards into stacks of identical time cycles, the operation, to examine T and to discard the card if it belonged to the previous time-cycle, was omitted here. To continue the calculation, the number λ^* is obtained by another random process; It is the (negative) logarithm of a random number which is equidistributed between 0 and 1. λ^* is then stored in its assigned accumulator and later, in the course of the computation used to determine

the distance to the next collision; It will be used to define the product of that distance with the total collision cross-section of the zone in question. (For obtaining and storing λ^* on Monte Carlo Flow Diagram I, see operation boxes: 1°, 2°, 3° and 4°.)

By using the given or, in the case of a fission neutron, the randomly chosen velocity, we then located the appropriate velocity interval, $0 \leqq k \leqq 9$ among the ten representative intervals. This was done by matching the V of the neutron to the V_k, the value of the ten velocity interval limits which have been set on the numerical function table at the beginning of the run. Having determined k, it is stored in its assigned memory location to be used later to find the appropriate cross-section of that velocity interval within the material zone in which the neutron under consideration is now travelling. (For determining and storing k, see Monte Carlo Flow Diagram I, operation boxes: $\overline{1}$, $\overline{2}$, $\overline{3}$, $\overline{4}$,, $\overline{5}$, $\overline{6}$ and $\overline{7}$.)

By using a, r and r_i, (r_i being the radii of the zones whose values were set on the numerical function table), we also determined the direction, ε of the neutron path, i.e. whether the particle was going in towards the center ($\varepsilon = 0$), or out towards the surface of the spherical assembly ($\varepsilon = 2$) and the distance, d, that it can travel from its present position, in its present direction before reaching the next zone boundary. (For computing and storing ε and d, see Monte Carlo Flow Diagram I, operation boxes: 18.0*, 18*, 19*, 20*, 21*, 22* and 23*.)

In the Second run, i.e. Monte Carlo Flow Diagram II, the operations to find λ^*, k, ε and d were, except for some storage location assignment, identical with the procedures as described above, however, the logical order in which these values were computed or found, was somewhat interchanged for the following reason: By interchanging the order obtaining

first ε and d, then k and lastly λ^*, we could in certain cases, shortcut and, thereby save machine computing time, in following the path of some neutrons. Thus, if $\alpha = 6$ or 7, indicating that the terminal event on the neutron card just read was census or zonal escape, V remains unchanged and its k can be directly read off the card. In the case of the undercritical assemblies, since all "source" cards (neutrons) were started at the center, by assigning the value of $d = r_1$, the total radius of the first zone, and $\varepsilon = 2$, the outward direction, we were able to omit in the initial calculation that part of the sequence where these quantities are being computed. Since the computation to obtain λ^* is common to all neutrons considered here therefore by leaving it last in the sequence in this part of the flow diagram, we could merge at this point of the calculation all branches previously separated by the value of α. (For obtaining ε, d, k and λ^* on Monte Carlo Flow Diagram II, see operation boxes: 20, … 26, 30, … 36 and 40, ….45).

As the next step in the computation, we found the material or materials with their appropriate densities, as present in the zone; f_i^b the density of material b in zone i. the total collision cross-section of the material present corresponding to the previously determined k velocity group was then formed, multiplied by f_i^b the material density of the zone.

For certain considerations of storage space on the numerical Function Table and, in order to be able to carry sufficient significant figures, scaled numbers S_J for the partial and Σ for the total cross-section, were used here instead of the true cross-section values. After having formed Σ it was multiplied by the appropriate scaling factor G in this manner obtaining σ, the true total cross-section. (The values for scaling

factors and the system of using them will be described in a separate section of the report.) The above described part of the computation, see: Monte Carlo Flow Diagram I, operation boxes $\overline{14}$, $\overline{15}$, $\overline{16}$, $\overline{17}$, and $\overline{17.1}$ for forming Σ and operation boxes 24.0*, 24.1*, 24.2* and 24* to find and apply the appropriate correction factor. However, on Monte Carlo Flow Diagram II, this whole calculation was included in one operation box (46), i.e. by "pre-mixing" through hand-calculation the cross-sections of all materials existing in the same zone, (we had allowed for four materials in each zone of the First run) we could omit the separate calculations for each b in that zone; this arrangement, and some further simplifications to be explained later, released sufficient space on the numerical Function table to store the true total cross-sections with sufficient significant numbers and already multiplied with the material density.

The true total cross-section, σ times the distance d was then compared with λ^*, the quantity that was introduced above. If $\sigma > \lambda^*$, we assumed that at the point $d_1 = {}^{\lambda^*}/_{\sigma}$ a collision occurred, which was temporarily treated as a fission, ($\alpha = 5$). If, however, $\lambda^* > d\sigma$, the neutron was presumed to have escaped, (temporarily noted as Total escape, $\alpha^* = 4$). Accordingly, in this case d was not modified, i.e. $d_1 = d$. For the neutron's path in the next zone the value of λ^* had to be reduced by the "risk" $d\sigma$ that it had already overcome, in order to keep the statistics unbiased. This means replacing λ^* by $\lambda^{**} = \lambda^* - d\sigma$. (See: Monte Carlo Flow Diagram I, operation boxes 25*, 26*, 27*, and Monte Carlo Flow Diagram II, operation boxes: 47, 48, 49.)

Next we examined whether there was indeed time enough (until the next census), for the neutron to cover (with its known velocity) the

distance d_1, and thus really reach the collision or, in case of escape, the limit of the zone. If covering the distance d_1 exceeds the census time T, the distance d_2 was formed for the point where the neutron arrived at census time, $d_2 = (T - t)V$, and t, the time of the event occurring was put equal to census time, $t = T$ and the terminal event was noted as census, $\alpha^* = 6$. If, however, while covering the distance d_1, t failed to exceed T, a new value $t^* = t + {}^{d_2}/_{V}$ was formed where $d_1 = d_2$. Merging the logical path for all neutrons, having suffered a collision, escape or census, we compute at this point the new values for a^* and r^*, since obviously any of the events will change the values of these quantities. (See: Monte Carlo Flow Diagram I, operation boxes: 28*, 29*, 29.1*, 29.2*, 30*, and Monte Carlo Flow Diagram II, operation boxes: 50, 51, 53, 55°, 56′, 54.)

In Monte Carlo Run I and II the logical sequence and actual computation, in this part of the flow diagrams, were essentially identical, except for the choice and use in the Second run of the value of t, which was carried here as a seven-place number, allowing three digits for the integer and four for the decimal part. As mentioned earlier in this report, it was desired in this run to make, if possible continuous computation, without having to stop the machine in order to permit the manual resetting of the value of Tat the end of each census cycle, (i.e., when all neutron cards still "alive" in the assembly, have reached time T), and to be able to feed cards back into the "Reader" of the machine without having to separate them, by outside sorting, into census stacks. On the other hand, it was desired, however, to note on each neutron card how many times it or its ancestors have passed through a census time during the course of the whole computation. In order to achieve these requirements, T, the census time unit, was replaced

by t_i the integer part of t. It was increased by one each time the terminal event, of the neutron under consideration, was determined to be census while t_d, the decimal part of t, was treated in the same way as in the First run. It was used to keep track of the time changes within a census cycle. t was, however, used in the Second run as a negative number, i.e. starting the calculation of a problem at $t = -x.0000$, where x could be chosen according to how many census cycles were desired before t_i reaching 0, would stop the calculation. Census time was started with $t_d = 0000$ and ended when $t_d \geqq .8192$. (It was found convenient here to take, as the end of a census cycle, 8192, the 13th [5] power of 2, since it facilitated the application of repeated halving factors to the γ of the "virgin" input cards as used in the under-critical assembly problems.)

Continuing now the course of the computation, we then examined α^* again to determine which path the particle will follow. If α^* was census, i.e. $\alpha^* = 6$, we gave q^* the appropriate weight: $q^* = 1$ in the First run where we manually duplicated or reduced each census stack to keep it an even hundred at the beginning of each census cycle, and $q^* = 2$ in the Second run where we automatically doubled the weight of each neutron which "survived" to the end of a census. g the fission generation, remained in this case unchanged from its previous value, i.e., since no fission occurred, the neutron was still within the same generation at which the particular card had started

[5] It was assumed that for the problems under consideration 13 census cycles would be sufficient.

at the beginning of this computation. (In view of the added need for storage space both on the IBM cards and on the machine and, since with the help of n', the "parent" number, it was possible to trace back neutron generations if this was necessary during the analysis of the results, g and for similar reasons n^*, the original "ancestor" numbers were both omitted in the Second run.)

The appropriate values are then rounded off to five significant figures and all values not yet in position are put in their proper printing locations, i.e. in the eight accumulators which are wired to transfer the numerical data from the ENIAC by printing them on IBM cards. At this point, after printing a card $\bar{n}$, the serial number for each printed card is increased by one and the q of the last read neutron card, i.e., the card for which a complete path has just been followed, as described above, is re-examined. Since this card could have been either census, fission, or, in the case of the Second run, zonal escape card,[6] the value of q can be any integer $\geqq 1 \leqq 4$. If $q = 1$, then the next card in the reader stack will automatically be read, starting the computation of the path of a new particle. This is achieved with the unconditional transfer (I) instructing the machine to recommence at the beginning of the local sequence. Since in the Second run a different routine was used at the initial steps of the "virgin" neutron cards of the under-critical assemblies, therefore, until computation on all of these has been finished, the unconditional transfer (A) was manually substituted to be used for the same purpose as (I). If, on the other hand, $q \geqq 2$, the path of each particle produced by fission will be separately followed, recommencing the computation with an unconditional transfer (K) at the

[6] The different treatments of zonal escape in the two Monte Carlo runs will be explained further down in this report.

point where a new random V and a are assigned to the fission neutron at the given r, i.e. the place where the fission, which produced this neutron, occurred. Before restarting the computation q is reduced by one, thus after having computed the path of the last fission neutron, q becomes 1 and a new card can be read from the stack. Since in the Second run census cards and, even in some cases, zonal escape cards had $q = 2$, the unconditional transfer (K) was directed to an earlier point at the beginning of the computation, i.e. to the place where fission cards were separated from census and zonal escape cards, as in the latter cases both branches of the "parent" neutron were started with the identical initial data. (See: Monte Carlo Flow-Diagram I, operation boxes: 31*, 37*, 37.1*, 47*, 48*, 49*, 50* and Monte Carlo Flow-Diagram II, operation boxes: 55, 64, 58, 58*, 58°, 58°, 59, 60, 61, 62, 63).

If the event was escape, $\alpha^* = 4$, we examined with the help of i, the present zone, ε, the direction of travel (in or out) and I, the total number of zones of the assembly under consideration, whether it was zonal or total escape. Total escape causes the same procedure as a census, i.e. it joins at this point the path preliminary to printing as described above. In the Second run, by putting $q^* = 0$ in the case of Total escape, an automatic removal from the Input stack, for those cards which should not be read again, has been facilitated (see operation box 10). (For Total escape see: Monte Carlo Flow Diagram II, operation boxes 32*, 34*, 35*, and Monte Carlo Flow Diagram II, operation boxes: 55*, 56, 57).

The event of zonal escape was treated in the First and Second run considerably differently: In the First run no card was printed if zonal escape occurred, but, after inspecting ε, thus determining and noting

whether the neutron passed into the inner or into the outer neighboring zone, the computation was shifted back, with help of an unconditional transfer order, to the point in the sequence (box 18.0*) where the direction and distance are being determined. Since no collision occurred, V remains unchanged and the total cross-section for the material in this zone is found by using the same k as in the previous calculation. λ^* has already been adjusted as noted earlier. The computation is then carried out as previously described until the next event has been determined and the point of the corresponding discrimination reached.

In the Second run a card was printed in the event of zonal escape indicating with the help of two additional symbols for α which type of zonal escape has occurred. In the Second run the use of a somewhat more detailed treatment of zonal escape was found necessary for the following consideration: Some of the problems in this run were computations on spherical assemblies consisting of 2 or 3 material zones, the outer zone (tamper) having disproportionally large dimensions from the point of view of calculation, i.e. the statistical information obtained through these calculations might have been rendered unwieldy by the size of the outer region. Also because of the very large scattering cross-sections in the tamper, relatively little information regarding other types of collisions could have been obtained between census-cycles in this region. Therefore, for this type of problems, zones of the same material compositions were divided into several zones of varying width and a weighting system was devised in the following manner: Two additional symbols for α^* were introduced to indicate the nature of the zonal escape, if after examining ε it was determined that the neutron particle was travelling "in" towards the center, its weight was doubled, ($q^* = 2$), and $\alpha^* = 7$ noted the event as zonal escape. If, on the other hand, the neutron was found to be travelling "out" towards the

surface of the assembly, a game of chance was played to determine whether or not after the present event the path of the neutron under consideration should be followed further. If the game was "won", α^* again becomes 7, but $q^* = 1$ – if the game was lost $\alpha^* = 2$, indicating a neutron whose genealogy was ended with this zonal escape and $q^* = 0$ for the same reasons as mentioned in the case of total escape. In all cases of zonal escape, we also examined whether the value of i_m, the zone most distant from the center in which the particle has ever travelled, will have to be substituted by a new value or remain unchanged. After this the path was merged with those of census and Total escape and a card was printed for each neutron having zonal escape as terminal events (See: Monte Carlo Flow Diagram II, operation boxes: 79 …. 85.)

Next, we follow the path were $\alpha^* = 5$, which indicates that a collision had occurred. We determine by a new random procedure what the specific character of the collision suffered by the neutron under consideration should be. This is done in the following manner:

The scaled total cross-section, Σ is taken negatively and multiplied with a random number equidistributed between 0 and 1. The result is g. To this we add S_0, the scaled cross-section of elastic scattering, multiplied with the density, f_i^b of the first material in this zone. (The values for f_i^b have been set on the Numerical Function table in ascending order of b for each zone.) The result g_1 is then the first alternative of the four collisions to be considered. If g_1 is positive, that is, if an "overdraft" has already occurred, then this collision is decreed to have been elastic scattering. If, on the other hand, g_1 is still negative, then the procedure is repeated with the other successive alternatives in this order: Inelastic scattering, fission, absorption. The corresponding scaled partial

cross-sections are S_1, S_2, S_3 and the successive results are g_2, g_3, g_4. If g_4 is still negative, this indicates that the first material was not responsible for the collision. In this case the procedure is repeated with the second, third and fourth material, (four being the maximum number of materials allotted to each zone in these computations), until an "overdraft" occurs, i.e. until the random decision for the type of collision has been obtained. Before, however, reiterating this procedure for any of the four materials, a preliminary test of its f_i^b is made: f_i^b was set with a negative value, (-1) on the Numerical Function Table whenever the corresponding material was not present in the zone under consideration. The sign discrimination on f_i^b causes, for a negative f_i^b , the omission of the detailed calculation, as described above, for this material and immediately the next material will be examined. (N.B. similar procedure has been already used earlier in the calculations at the point where the sum of the total cross-section in a particular zone was being formed; see: Monte Carlo Flow Diagram I, operation box $\overline{15}$.)

In Monte Carlo Flow Diagram II some changes, i.e. simplifications, have been made in this section of the calculation, retaining, however, the basic method of the random decision to determine the type of collision occurred. As mentioned earlier in this description, for the Second run various materials existing in the same zone were "pre-mixed" and set on the Numerical Function Table already weighted by the appropriate material densities. This permitted then the omission of the use of f_i^b in the machine calculation and consequently shortened the calculation by having had to examine only one material for each zone in order to determine the collision path. Similarly in the

Second run, fission and absorption cross-sections have been also "pre-mixed" by manual calculations with the effect that only two tries had to be made to determine which type of collision has been decreed. The relative values of Elastic and Inelastic scattering, $(^{\sigma_E}/_{\sigma_t}, ^{\sigma_I}/_{\sigma_t})$ are set on the Numerical Function Table. If after adding these to the negative random number, no "overdraft" occurred, no further try was necessary and the calculation was then made to follow the fission-absorption path. (For this part of the calculation see: Monte Carlo Flow Diagram I, operation box: $\overline{18}$, $\overline{18.1}$, …. $\overline{27}$, and Monte Carlo Flow Diagram II, operation box: 65, … 69.)

Following the path of the positive branch of g_1 – the path the neutron would travel, if by the above-described random method the collision had been decreed to be Elastic scattering, a new value for a was calculated using the same random method as described earlier. Provisions were also made to calculate a new value of V in those cases where Elastic scattering occurred in light material. In the First run neither of the six assemblies under consideration contained any light material. In the Second run, Elastic scattering in light material was treated as an added, separate collision-path, g_3 computing a new value of V for the elastically-scattered particle only if the "overdraft" occurred at this point.[7] (Not shown on Monte Carlo Flow Diagram II.)

If the "overdraft" occurred at g_2 i.e., the collision was decreed to be Inelastic scattering, a new value for V was computed [5] and the path merged with that of Elastic scattering at the point where the computation to obtain the new value of a commenced. After this operation,

[7] For explanation of computing the new value of V at Elastic scattering in light material and in the case of Inelastic scattering, see chapter on physics, p. ____.

an added step was done in the Second run in order to increase by one the count S representing the number of scatterings suffered by the card, (neutron) under consideration. Then, through the means of an unconditional transfer, the computation was shifted back to that part of the logical sequence where the calculations for the new values of ε, d, λ and, if necessary, k are effected. The calculation is then followed on its path until an event other than scattering, i.e. census, escape or fission-absorption, has occurred. If the event is census or escape, the corresponding path, (described on the previous pages), will be followed in the computation. If, however, in the event of collision, fission occurred, a game of chance was played to determine the number of neutrons produced by this collision. The appropriate probability for one, two or three-way fission in the material under consideration was set on the Numerical Function Table. This compared to a random number equidistributed between 0 and 1 made the decision to choose the value of q^* as the appropriate weight which this neutron card, when again read, will represent. (In the First run, p_q^b represented the one, two or three-way fission-probability; in the Second run q^* was chosen as ν_{Int} or as $\nu_{Int} + 1$, according to the decision made by comparing ν_{dec} with the random number. In the case of the Second run q^* could also be 0, since here the fission probabilities were reduced or, as in some materials, disappeared due to having "pre-mixed" by hand calc[ulation] the fission and absorption cross-sections.) At this point, in the First run g the generation index, was increased by one; this family identification number was however, omitted in the Second run, for the reasons as mentioned earlier.

Since in the First run, Absorption was treated as a separate collision type which happened in the case when the "overdraft" occurred on g_4 following then the positive branch of this sign discrimination, α^* was put equal to 3, thereby noting that Absorption was the terminal event for the neutron under consideration. Both Fission and Absorption branches then joined the point preparatory to printing and from here on followed the common path of the other, (census, escape), terminal events. (For treatment of collisions, see Monte Carlo Flow Diagram I, operation boxes: 51.1*, 51*, 52*, 53*, 54*, 38*, 39*, 46*, 36*, and Monte Carlo Flow Diagram II, operation boxes: 70, …. 73.)

In the above description we have attempted to explain the meaning and use of the various paths of the logical sequence as shown on the Flow Diagrams of the First and Second runs of Monte Carlo type calculations on the ENIAC. To conclude the description of the Flow Diagrams, the method applied to obtain and "refresh" the random numbers used throughout the computation at various points is given here as follows: A randomly chosen n digit number ξ was taken as the base ($n = 8$ in the First run, $n = 10$ in the Second run), squared and the middle n digits used as, on one hand, the next supply of random digits, as needed in the course of the computation and, on the other hand, as the base of the next number to be squared. This "refresh" method could have been repeated as many times as the calculations in the main part of the logical sequence required it. Since, however, in the first case, only two thousand and in the second case, little over three thousand squarings were statistically tested for randomness, provisions were made to recommence with the original random numbers each time when, in

the course of the calculations two (three) thousand squarings have already been made. Having reached this point, the machine was instructed to read a "special" card, containing the original random number and to place it into the accumulator assigned as, more or less, permanent memory location for the changing value of ξ. This "special" card, as used in the First run, also carried $\alpha = 7$ and $\bar{n}$, the serial number; $\alpha = 7$ was needed to separate the path of this card from those which were normally used in the course of the computation, $\bar{n}$ was used to retain intact the neutron card count. In the Second run, a count order, devised specially for this purpose, instructed the machine to stop after the appropriate number of squaring. Then ξ was replaced in its permanent memory location by the original random number changing, however, by appropriate shift orders, the order in which the digits were used to determine the various random paths for the neutron to travel. (This method was used in order to add to the randomness of the calculation.)

Although lists of the checked random numbers had been available at the start of either run, it was found more convenient and faster to have the squaring, i.e. "refresh" done by the ENIAC. The "refresh" random number is shown on a separate block on the flow-diagram, with (ρ) indicating the places where a new random number was needed and (ω_n) the various exits to the particular path where the computation had to be continued after obtaining the new random number.