TRI-77-4

TRIUMF

MONTE CARLO SIMULATION OF EXPERIMENTS

An Introduction
Illustrated by the Computer Programs

SIMUL8

($\mu^+ \to e^+ \nu_e \bar{\nu}_\mu \gamma$ in TINA and MINA NaI Detectors)

MONTE-PION

($\pi^+ \to e^+ \nu_e \gamma$ in TINA and MINA NaI Detectors)

Geoffrey I. Opat

Visitor to
TRIUMF and Physics Department
University of British Columbia

On leave from
School of Physics
University of Melbourne

Postal address:

TRIUMF
University of British Columbia
Vancouver, B.C.
Canada V6T 1W5

July 1977

ABSTRACT

An outline of the technique of computer simulation of particle physics experiments by the MONTE-CARLO method is presented. Useful special purpose subprograms are listed and described. At each stage the discussion is made concrete by direct reference to the programs SIMUL8 and its variant MONTE-PION, written to assist in the analysis of the radiative decay experiments $\mu^+ \to e^+ \nu_e \bar{\nu}_\mu \gamma$ and $\pi^+ \to e^+ \nu_e \gamma$, respectively. These experiments were based on the use of two large sodium iodide crystals, TINA and MINA, as e and γ detectors. Instructions for the use of SIMUL8 and MONTE-PION are given.

I. INTRODUCTION

The following reactions have been, or will be, investigated at TRIUMF using the large sodium iodide detectors TINA and MINA:-

$$\pi^- p \rightarrow \underset{\llcorner\!\!\longrightarrow \gamma\gamma}{\pi^\circ n} \tag{1}$$

$$\pi^+ \rightarrow \underset{\llcorner\!\!\longrightarrow \gamma\gamma}{\pi^\circ e^+ \nu} \tag{2}$$

$$\pi^+ \rightarrow e^+ \nu \gamma \tag{3}$$

$$\mu^+ \rightarrow e^+ \nu \bar{\nu} \gamma \tag{4}$$

$$\mu^+ \rightarrow e^+ \gamma \tag{5}$$

All these reactions result in two energetic particles; either two gamma rays, or an electron and a gamma ray. With the initial particle being at rest, the final configurations of interest have the momenta of the two most energetic particles almost antiparallel.

Reactions 2-5 are accompanied by normal muon decays

$$\mu^+ \rightarrow e^+ \nu \bar{\nu} \tag{6}$$

Reaction 6, having a single energetic detectable particle is recorded as a "singles" event. Reactions 1-5 having two such particles are recorded as "doubles" events.

Consider for example reactions 4 and 5 which were investigated with the apparatus shown in figure 1. Scintillation counters are denoted by the symbols #1, #2, ..., #10. Collimator apertures are denoted by #20, ..., #23. A degraded (and hence multiply-scattered) pion beam is stopped by counter #3, the stop signal being $1\cdot2\cdot3\cdot\overline{4}$. The μ^+ is produced by the decay $\pi^+ \rightarrow \mu^+\nu$ which gives a delayed pulse in #3. An electron is recorded as having entered TINA by the logic signal $5\cdot7\cdot9\cdot$(TINA). A gamma ray is recorded as having entered MINA by the signal $\overline{6}\cdot\overline{8}\cdot\overline{10}\cdot$(MINA). Doubles events with the roles of TINA and MINA interchanged have a suitably modified logic.

Whereas the logic signals assist in identifying the events and loosely restricting their geometric configuration, the pulse-height analogue signals from TINA and MINA are the only truly kinematic variables recorded. These pulse-heights are related to the electron and gamma ray energies E_e and E_γ respectively, by the response functions of TINA and MINA. A typical response function is sketched in figure 2. (See Appendix I for a detailed discussion.)

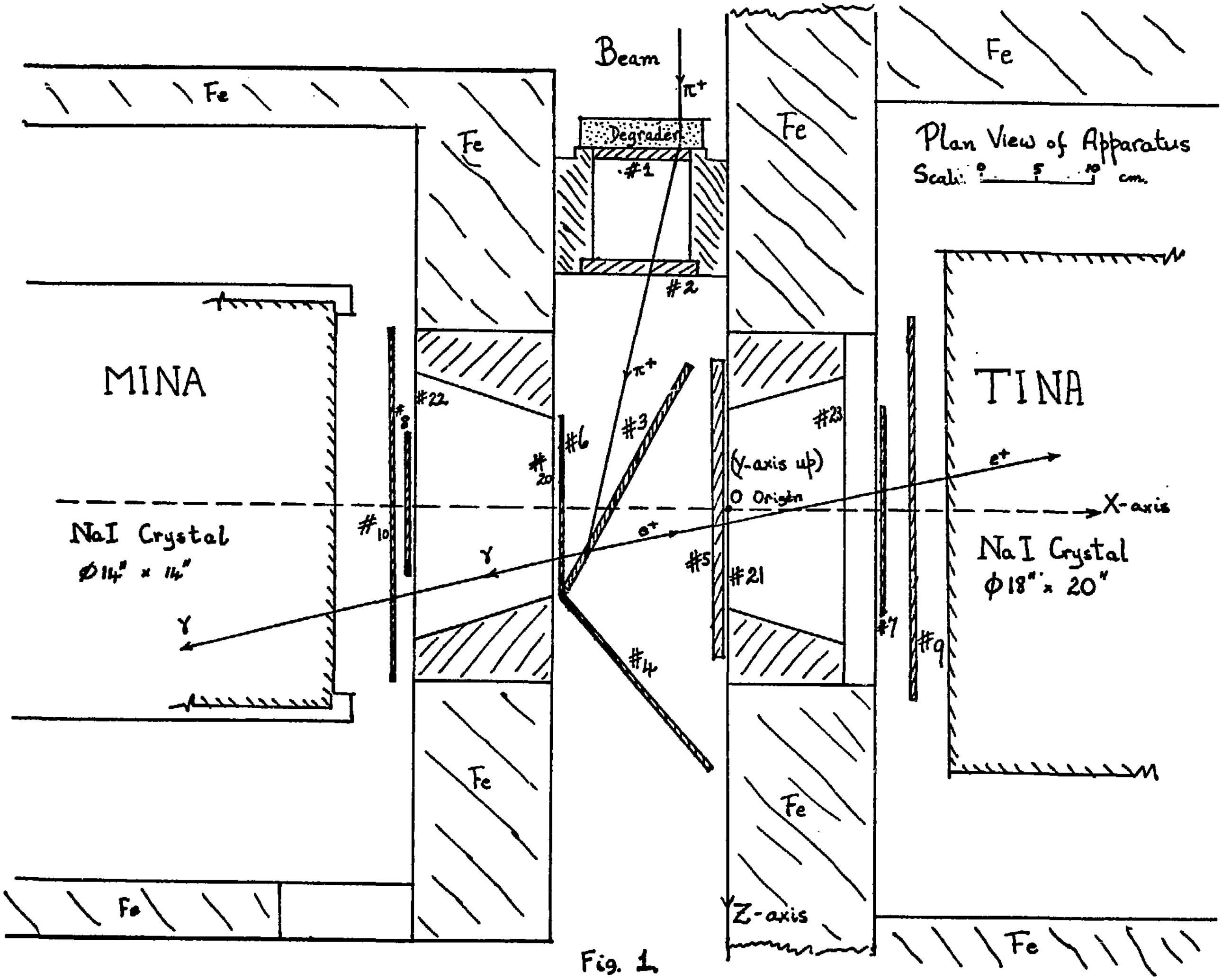

Beam
π^+
Degrader
#1
#2
Fe
Fe
Fe
Fe
Fe
Fe
Fe
Fe
Plan View of Apparatus
Scale: 0 5 10 cm.
MINA
NaI Crystal
Ø14" × 14"
TINA
NaI Crystal
Ø18" × 20"
π^+
#3
#4
#5
#6
#7
#8
#9
#10
#20
#21
#22
#23
(Y-axis up)
O Origin
X-axis
Z-axis
e^+
e^+
γ
γ

Fig. 1.

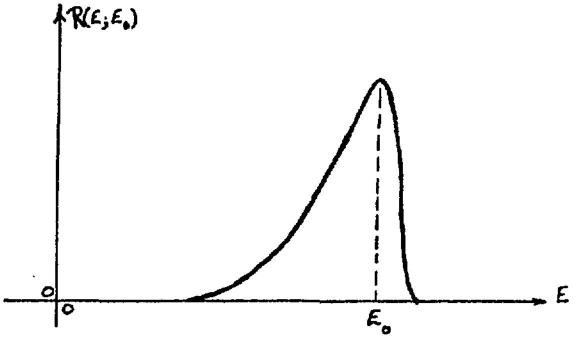


Fig. 2. A calibrated Sodium Iodide detector response function. $R(E,E_0)dE$ is the probability that a particle of energy E_0 will give a pulse height corresponding to the range $(E,E+dE)$.

The angle between the electron and gamma ray directions $\theta_{e\gamma}$ is kinematically restricted by the requirement that the invariant mass of the undetected neutrino pair be positive. Specifically

$$\mathrm{Max}(0,(x+y-1)/xy) \lesssim \mathrm{Sin}^2(\tfrac{1}{2}\theta_{e\gamma}) \lesssim 1 \qquad 7$$

where*

$$x = E_e/E_{max} \qquad 8a$$

$$y = E_\gamma/E_{max} \qquad 8b$$

$$E_{max} = \tfrac{1}{2} M_\mu \qquad 8c$$

The kinematic situation is shown in figure 3.

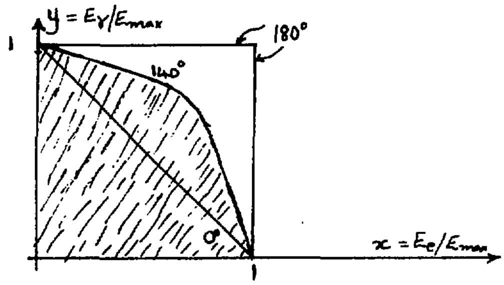


Fig. 3. The kinematically allowed region of (x,y) for a given $\theta_{e\gamma}$. The labels on the curves are $\theta_{e\gamma}$. The allowed region for $\theta_{e\gamma}$ = 140° is shaded in.

*Some authors interchange these definitions of x and y.

Events from the putative decay $\mu^+\to e^+\gamma$ (reaction 5) only populate the point $(x,y) = (1,1)$. Events from radiative muon decay (reaction 4) are dense near the point $(0,0)$ and along the line $y = 0$, and are sparse near the point $(1,1)$. In fact the number of events in the interval $1-\varepsilon \gtrsim x,y \gtrsim 1$ is proportional to ε^6. In addition, however, the finite widths of the TINA and MINA energy response functions "smear" the x-y plane distributions significantly, and thus limit the separability of events from reactions 4 and 5.

If the objectives of the experiment outlined above are (i) to measure the $\mu^+\to e^+\gamma$ branching ratio and (ii) to improve our knowledge of the radiative muon decay (branching ratio and η' parameter), what information would an experimenter want to have to assist in the planning and data analysis stages of the work?

A suggested list would include:

(a) Singles detection probability per stop, i.e. effective solid angles.

(b) Singles energy distribution, i.e. a Michel spectrum convoluted with the energy response function.

(c) Doubles detection probability for each region of the (x,y) plane. (As can be seen from figure 3, each point in the x-y plane is associated with a range of angles $\theta_{e\gamma}$. Given the extended target (#3) and the disposition of the counters (#7, #8) and the apertures (#20, ..., #23) the degree of acceptance of events into TINA and MINA is far from simple to compute. This is true even close to the point (1,1) where the e and γ are antiparallel).

(d) The x distribution and y distribution of the (cut) x-y plane.

(e) The sensitivity of the above distributions to geometric misalignments of the apparatus.

(f) The sensitivity of the above distributions to finite energy resolution.

(g) The sensitivity of the above distributions to beam distribution on the target.

(h) Assessment of background rates from such reactions as $\pi^+\to e^+\nu\gamma$, $\pi^+\to\underset{\llcorner\!\longrightarrow\gamma\gamma}{\pi^\circ e^+\nu}$. (A good assessment helps in the recognition of backgrounds.)

This is what the programs SIMUL8 and MONTE-PION are about. However, first we shall need a little theory.

II. Event Detection Probability Formulated Mathematically

We consider here the detection of radiative muon decay events (reaction 4) by the apparatus of figure 1. We seek to compute the probability, w, that we detect the radiative decay of a single muon, given a set of "cuts" on the energy variables x,y. Let Γ be the total muon decay rate. Let $d\Gamma_\gamma$ be the differential radiative decay rate. The differential branching ratio, db, is given by

$$db = \frac{d\Gamma_\gamma}{\Gamma} \equiv f(\vec{e},\vec{\gamma}) \cdot \frac{d\Omega_e}{4\pi} \cdot \frac{d\Omega_\gamma}{4\pi} \cdot dx \cdot dy \qquad 9$$

$f(\vec{e},\vec{\gamma})$ is a function of the electron and gamma ray momenta $\vec{e}$ and $\vec{\gamma}$ given by some theoretical expression, (function subprogram THEORY in fact). db is a differential of the 6th order.

Let dP_B be the differential probability that a beam particle will emerge from a certain point of the degrader into a given direction and penetrate a thickness t of #3 before stopping in #3. (dP_B is a differential element of the 5th order. The distribution associated with this element is generated by BEAMIN.)

Let dP_{Ei} be the probability that the counter i gives out pulse height energy x' when an energy x is deposited.

$$\begin{aligned} &i = \text{TINA or MINA, or T and M} \\ &\text{i.e. } dP_{Ei} = R_i(x,x')dx' \end{aligned} \qquad 10$$

Let $a_j(e)$ be an aperture function for the j^{th} aperture

$$\begin{aligned} a_j(e) &= 1 \quad \text{if e goes through \#j} \\ &= 0 \quad \text{if e does not go through \#j} \end{aligned} \qquad 11$$

Let $C(i,x',y')$ be a cut function corresponding to variables i = T or M; x',y' lying in some specified range

$$\begin{aligned} C &= 1 \quad \text{if variables are inside cut} \\ &= 0 \quad \text{if variables are outside cut} \end{aligned} \qquad 12$$

With the aid of figure 1 we write for w

$$\begin{aligned} w = \int db.dP_B\Big\{&a_7(e)a_{21}(e)a_{20}(\gamma)a_{22}(\gamma)R_T(x,x')R_M(y,y')C(T,x'y')dx'dy' + \\ &a_8(e)a_{20}(e)a_{21}(\gamma)a_{23}(\gamma)R_M(x,x')R_T(y,y')C(M,x'y')dx'dy'\Big\} \end{aligned} \qquad 13$$

We note in passing that eq. 12 contains a 13-dimensional integration with quite a complex boundary.

In essence (but not detail) eq. 13 has the form

$$w = \int f(\overline{x})\,dp \qquad 14$$

where dp is a multidimensional element of probability. We turn to the investigation of the evaluation of such forms by the MONTE CARLO method.

III. General Theory of the Monte-Carlo Method

Consider the integral

$$\mu \equiv \int f(\vec{x})P(\vec{x})d^n x = \int f(\vec{x})\,dp \qquad 15$$

In this expression $f(\vec{x})$ is a given function called the weight function. $P(\vec{x})$ is a given function with the properties

$$P(\vec{x}) \geq 0 \qquad 16a$$

$$\int P(\vec{x})d^n x = 1 \qquad 16b$$

We can think of $P(\vec{x})$ in several different ways:

i) $dp = P(\vec{x})d^n x$ represents the probability that a point is to be found in the range $d^n x$ at $\vec{x}$. In the present work this is how we shall interpret the function $P(\vec{x})$.

ii) $P(\vec{x})$ is the Jacobian of a co-ordinate transformation.

iii) $P(\vec{x})$ is just simply a positive, normalised fixed function.

iv) Sometimes $P(\vec{x}) = {}^1/_V$ = constant, V = volume of integration.

We suppose that we have at our disposal (as a computer program) a random point generator which generates the n-tuples $(x_1,x_2,..,x_n)$ such that the probability dp of finding a point in the range $d^n x$ at $\vec{x}$ is $P(\vec{x})d^n x$.

We consider two statistics S_1, S_2 formed from N points drawn from the random point generator. Explicitly

$$S_1 \equiv \frac{1}{N}\sum_{i=1}^{N} f(\vec{x}_i) \qquad 17$$

$$S_2 \equiv \frac{1}{N-1}\left\{\frac{1}{N}\sum_{i=1}^{N} f^2(\vec{x}_i) - \left(\frac{1}{N}\sum_{i=1}^{N} f(\vec{x}_i)\right)^2\right\} \qquad 18a$$

$$= \frac{1}{N-1}\left\{\frac{1}{N}\sum_{i=1}^{N} f^2(\vec{x}_i) - S_1{}^2\right\} \qquad 18b$$

Before estimating the values of S_1 and S_2 we define the mean and variance of the function $f(\vec{x})$ by:

$$\mu \equiv \int f(\vec{x})P(\vec{x})d^n x \qquad 19a$$

$$\sigma^2 \equiv \int (f(\vec{x})-\mu)^2 P(\vec{x})d^n x \equiv \int f^2(\vec{x})P(\vec{x})d^n x - \mu^2 \qquad 19b$$

Consider the expectation values

$$\langle S_1 \rangle = \frac{1}{N}\sum_{i=1}^{N} \langle f(x_i) \rangle = \langle f(x) \rangle = \int f(\vec{x})P(\vec{x})d^2 x = \mu \qquad 20$$

$$\langle (S_1-\mu)^2 \rangle \equiv \langle S_1{}^2 \rangle - \mu^2 = \frac{1}{N^2} \sum_{i,j=1}^{N} \langle f(x_i)f(x_j) \rangle - \mu^2$$

$$= \frac{1}{N^2}\left\{ N \langle f^2(x) \rangle + N(N-1) \langle f(x) \rangle^2 \right\} - \mu^2 = \sigma^2/N \qquad 21$$

Thus the statistic S_1 has mean μ, and standard deviation $\sigma/\surd N$. Equation 17 exhibits S_1 as the sum of a (large) number of independent terms, and therefore (by the central limit theorem) we conclude that S_1 has a normal distribution (of known mean and variance).

Notes i) If N is large and σ is small it is highly improbable that S_1 will deviate from μ appreciably. We therefore regard S_1 as the computed value of μ with "error", ± $\sigma/\surd N$.

ii) In general we do not know σ for the function $f(\vec{x})$ and thus we do not know how well we have determined μ. We now turn to statistic S_2 for assistance with this problem.

Consider the expectation value of statistic S_2 of eq. 18b.

$$\langle S_2 \rangle = \frac{1}{N-1}\left\{ \frac{1}{N}\sum_{i=1}^{N} \langle f^2(x_i) \rangle - \langle S_1{}^2 \rangle \right\} = \frac{1}{N-1}\left\{ \mu^2+\sigma^2 - \langle S_1{}^2 \rangle \right\} = \sigma^2/N \qquad 22$$

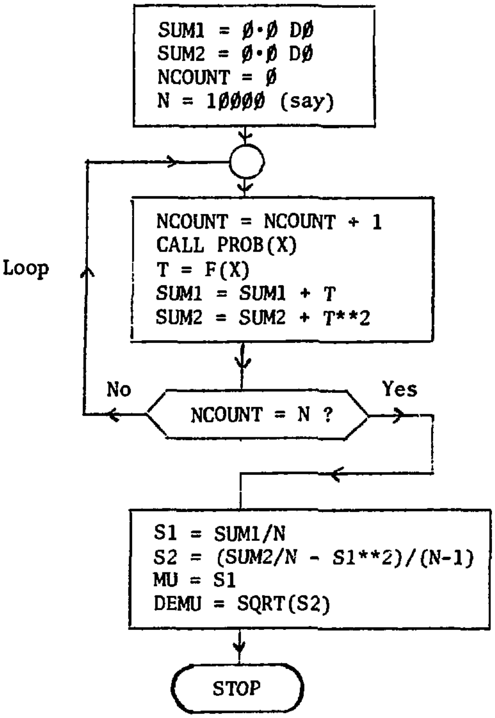
The last equality comes from equation 21 in which it is shown *(inter alia)*.

$$\langle S_1{}^2 \rangle - \mu^2 = \sigma^2/N$$

Summary $S_1 \pm \sqrt{S_2}$ is regarded as the computed value and error of μ, in the sense of a normal distribution.

Note 1 As the results show, the computed accuracy of the integral of eq. 1 does not depend on the dimensionality, n, of the space of integration. This fact underlies the reason for the importance of Monte-Carlo methods in multi-dimensional problems.

Note 2 A computer program looks like:



```
SUM1 = Ø·Ø DØ
SUM2 = Ø·Ø DØ
NCOUNT = Ø
N = 1ØØØØ (say)
```

```
NCOUNT = NCOUNT + 1
CALL PROB(X)
T = F(X)
SUM1 = SUM1 + T
SUM2 = SUM2 + T**2
```

```
S1 = SUM1/N
S2 = (SUM2/N - S1**2)/(N-1)
MU = S1
DEMU = SQRT(S2)
```

Notes: (1) SUM1, SUM2, S1, S2 } All double precision

(2) Loop time must be kept to a minimum

Fig. 4

IV. Technical Matters

We now turn to four problems of MONTE-CARLO programs. These are, random distribution generators, specification of integration boundaries, cuts or subintegrals, and variance reduction.

(i) Random Distribution Generators

The most important random number generator is a computer function which generates a distribution uniform in the interval (0,1). i.e. if x_n is the value of the function on the n^{th} call, 0<x<1.0, and x_n has probability dx of falling in the interval (x,x+dx). The successive values x_n,x_{n+1} are correlated. The FORTRAN compiler at the

UBC IBM installation recognizes the statement:

$$X = \text{FRAND}(0\cdot0)$$

The argument of FRAND is dummy. The sequence $\{X_n\}$ is determinate, but can be varied by one initial call

$$Z = \text{RAND}(Y)$$

The value of the real, Y, determines the sequence $\{X_n\}$ that will be produced by FRAND. If a different sequence $\{X_n\}$ is required on successive computer runs, the value of Y should be changed. The FRAND call time seems to be about 9 μsec.

Although other random number generators are often supplied by the computer library, and are useful on occasion, the main programming task is to transform the uniform distribution in (0,1), into the multidimensional distribution $P(\vec{x})$.

ii) Integration Boundaries

In a multidimensional space, the region of integration may be easy to specify verbally, but impossibly difficult and/or time-consuming to specify analytically. For example (with reference to figure 1), the set of all event co-ordinates corresponding to a beam particle stopping in #3 with an electron traversing #21 and #7 and a gamma ray traversing #20 and #22, constitutes a point within the region of integration. The edge of the (13-dimensional!) region so specified defies analytic description. (Actually, another 13-dimensional region has to be added when the particles go into the opposing counters.)

The way around this problem is to replace the original problem with a related one.

Originally we had (in a simplified notation)

$$\mu = \int_r f(x)P_r(x)dx \qquad 23a$$

$$\sigma^2 = \int_r f^2(x)P_r(x)dx - \mu^2 \qquad 23b$$

where $$\int_r P_r(x)dx = 1 \qquad 23c$$

The region of integration is denoted by r. It is the one with an "evil" boundary.

Let R be a region with a "good" boundary such that $r \subset R$. Within R we extend the probability function to $P_R(x)$ in a way which satisfies

$$\int_R P_R(x)dx = 1 \qquad 24a$$

and

$$P_r(x) = \frac{1}{h} P_R(x), \qquad x \in r \subset R \qquad 24b$$

The normalization constant, h, is the probability that a point of R lies within r. We call h the "hit" probability i.e. chance a point of R hits r.

$$h = \int_r P_R(x)dx \qquad 25$$

We introduce also the "miss" probability m.

$$m = 1-h \qquad 26$$

Further, we introduce the boundary function $B_r(x)$.

$$\begin{aligned} B_r(x) &= 1 \qquad x \in r \\ &= 0 \qquad x \notin r \end{aligned} \qquad 27$$

Writing programs for such functions is one of the programmers tasks, it is usually not too difficult, as $B_r(x)$ can usually be represented as the product of several simple factors.

We introduce the new weight function

$$F(x) = f(x)B_r(x)/h \qquad 28$$

The new problem has mean M and variance Σ^2,

$$\begin{aligned} M &\equiv \int_R F(x)P_R(x)dx \\ &= \mu \qquad \text{(as before)} \end{aligned} \qquad 29$$

$$\begin{aligned} \Sigma^2 &\equiv \int_R (F(x)-M)^2 P_R(x)dx \\ &= (\sigma^2+m\mu^2)/h \qquad \text{(worse than before)} \end{aligned} \qquad 30$$

If h is small the increase in error may be intolerable unless N is increased significantly.

Problem The programmer has to decide whether it is better to waste the machine's time by doing a large N run, or his time by slaving over an analytic boundary specification (which may also be costly to compute). Whichever way things go, you cannot just go on increasing N. Sometimes the solution comes by hybridising the problem, in some directions in R-space specifying the boundary analytically, in other directions extending the boundary a little.

Shrinking Quite often it is the reverse problem which has to be tackled. Suppose the weight f(x) contains a factor b(x) which goes to zero in some domain, then one would want to shrink the region r down onto this domain to improve efficiency.

Case Studies from SIMUL8

(a) Analytic specification and shrinking

We need to generate a compatible set of variables x,y,Cos$\theta_{e\gamma}$ such that

$$x_l \leq x \leq x_u \qquad 31a$$

$$y_l \leq y \leq y_u \qquad 31b$$

$$\mathrm{Max}(0,(x+y-1)/xy) \equiv z_l \leq z \leq 1 \qquad 31c$$

where $$z \equiv \mathrm{Sin}^2 \tfrac{1}{2}\theta_{e\gamma} \qquad 31d$$

Let f(x,y,z) be zero whenever x,y,z lie outside the ranges specified by equations 31a, b, c. Equations 31 a, b, c define the region r.

$$\mu = \int_0^1 dx \int_0^1 dy \int_{-1}^1 \frac{d\mathrm{Cos}\theta_{e\gamma}}{2} f(xyz) \equiv \int_0^1 dx \int_0^1 dy \int_0^1 dz\ f(xyz) \qquad 32$$

$$= \int_0^1 dX \int_0^1 dY \int_0^1 dZ\ F(XYZ) \qquad 33$$

where $$F(xyz) = f(xyz)(x_u - x_l)\cdot(y_u - y_l)\cdot(1 - z_l(xy)). \qquad 34$$

With $$o \leq X,Y,Z \leq 1$$

and the relations 35 below we have analytically shrunk the space

$$\left\{o \leq x,y \leq 1;\ -1 \leq \mathrm{Cos}\theta_{e\gamma} \leq 1\right\} \equiv R$$

onto the space r.

$$x = x_l + (x_u - x_l)X \qquad 35a$$

$$y = y_l + (y_u - y_l)Y \qquad 35b$$

$$z = z_l(x,y) + (1 - z_l(x,y))Z \qquad 35c$$

(b) Hybridisation

The integral over electron and gamma ray directions occurs as a subpart of the program SIMUL8.

$$\text{i.e.} \qquad \mu = \int_{4\pi} \frac{d\Omega_\gamma}{4\pi} \int_{4\pi} \frac{d\Omega_e}{4\pi} \; f(\hat{e},\hat{\gamma}) \qquad 36$$

Where $f(\hat{e},\hat{\gamma})$ is defined as zero when (see figure 1) the electron does not traverse #7 and #21, and the gamma ray does not traverse #20 and #22. Obviously we are wasting points if we allow the electron direction $\hat{e}$ to be generated in all 4π directions as the chance of hitting #7 is small. We thus transform the integral to an integration over the area of the disk of #7.

$$\frac{d\Omega_e}{4\pi} = \frac{\hat{e}\cdot\hat{n}\;A}{4\pi r^2} \cdot \frac{dA}{A} \qquad 37$$

Where r is the distance from the interaction point to the electron entry point on #7, A is the area of the counter and $\hat{n}$ is its normal (which points away from the source of radiation). Thus the solid angle probability element $d\Omega_e/4\pi$ has been transformed into the areal probability element dA/A. i.e. dA/A is the chance an electron which does fall on A, falls inside dA.

$d\Omega_\gamma/4\pi$ is computed regarding the electron direction to be the "Z-axis".

$$\text{i.e.} \qquad \frac{d\Omega_\gamma}{4\pi} = \frac{d\phi_{e\gamma}}{2\pi} \cdot \frac{d\text{Cos}\theta_{e\gamma}}{2} \qquad 38$$

The probability element $d\text{Cos}\theta_{e\gamma}/2$ has been discussed in the previous section (a), the probability element $d\phi_{e\gamma}/2\pi$ was computed with the aid of RANDIR(CS,SN) which generates the sine and cosine of a random direction in the plane.

With these changes the new weight function is:

$$F(\hat{e},\hat{\gamma}) = \frac{\hat{e}\cdot\hat{n}\;A}{4\pi r^2} \cdot f(\hat{e},\hat{\gamma}) \qquad 39$$

In the region of energies $x,y \simeq 1$ the electron and gamma ray move almost antiparallel. With the electron constrained to traverse #7, we find the overall hit probability for the entire event to be $h \simeq 0.8$, only a small source of variance increase.

iii) Cuts and Slices

In most applications one wants to know, not only the overall detection rate, but also the detection rate corresponding to say a particle energy in some subrange. Such events are "cut" out as they occur in the MONTE CARLO loop of figure 4 and their subtotals are also formed. In fact, a very large number of one and two dimensional cuts are often performed simultaneously within the main loop in order to display the functional dependence on the selected variables.

From the point of view of a single cut it is equivalent to the replacement in eq. 15

$$f(x) \rightarrow f(x)C(x) \qquad 40$$

where

$$C(x) = 1 \qquad x \; \varepsilon \; c \subset r$$
$$\;\; = 0 \qquad d \; \varepsilon \; c \subset r \qquad 41$$

and c is the region of the cut within r.

Analogous considerations to those given previously show:

$$\mu_c = h\mu'_c \qquad 42a$$

$$\sigma^2{}_c = h(\sigma'^2_c + m\mu'^2_c) \qquad 42b$$

In eqs. 42, h and m are the cut hit and miss probabilities and μ'_c and σ'^2_c are the mean and variance of f within the cut. Quite often $\sigma'_c \simeq 0$ and $m \simeq 1$. In this case the fractional uncertainty in μ_c is given by

$$\frac{\sigma_c}{\mu_c\sqrt{N}} \simeq \frac{1}{\sqrt{hN}} \qquad 43$$

which is simply the expected Poisson partitioning error. In fact eq. 43 gives a minimum value to the error within a cut.

iv) Variance Reduction

It is clear that if the variance of f(x) is large the MONTE CARLO method will work poorly. Several devices are available to reduce the variance and this subject forms much of the content of the literature on the method (Refs. 1, 3). One device is simple if it can be applied.

Suppose a function $Q(x)$ can be found, such that $f(x)/Q(x)$ has a small variance. Then the replacement

$$f(x) \rightarrow f(x)/Q(x) = f'(x) \qquad \text{44a}$$

$$P(x) \rightarrow Q(x)P(x) = P'(x) \qquad \text{44b}$$

can often give a great improvement. The present program chains do not have explicit variance reduction included. This is partly because of the requirement of uniform (fractional) accuracy within each cut as well as good overall accuracy, and partly because of its difficulty of implementation.

<u>One dimensional example of problems (i) - (iv)</u>

We compute the trivial integral

$$\int_0^1 2x dx = 1$$

using the MONTE CARLO method.

$$S_1 = \frac{1}{100} \sum_{i=1}^{100} 2x_i \quad \text{with } 0 \leqslant x_i \leqslant 1$$

This process was repeated 1000 times using a different set of random numbers each time. The distribution of values of S_1 is shown in figure 5a. The mean and variance of the 1000 numbers so found is in good agreement with that expected from the theoretical evaluation of the variance.

In figure 5b the effect of extension of the boundary are manifest. Note to compute this case the following sum is found,

$$S_1 = \frac{1}{100} \sum_{i=1}^{100} 2(2x_i)\theta(1-x_i). \text{ with } 0 \leqslant x_i \leqslant 2$$

In figure 5c the variance reduction technique is illustrated. The variable was $y = \text{Cosh}x \in (1, \text{Cosh}(1))$.

In figure 5d the larger variance within a cut is illustrated.

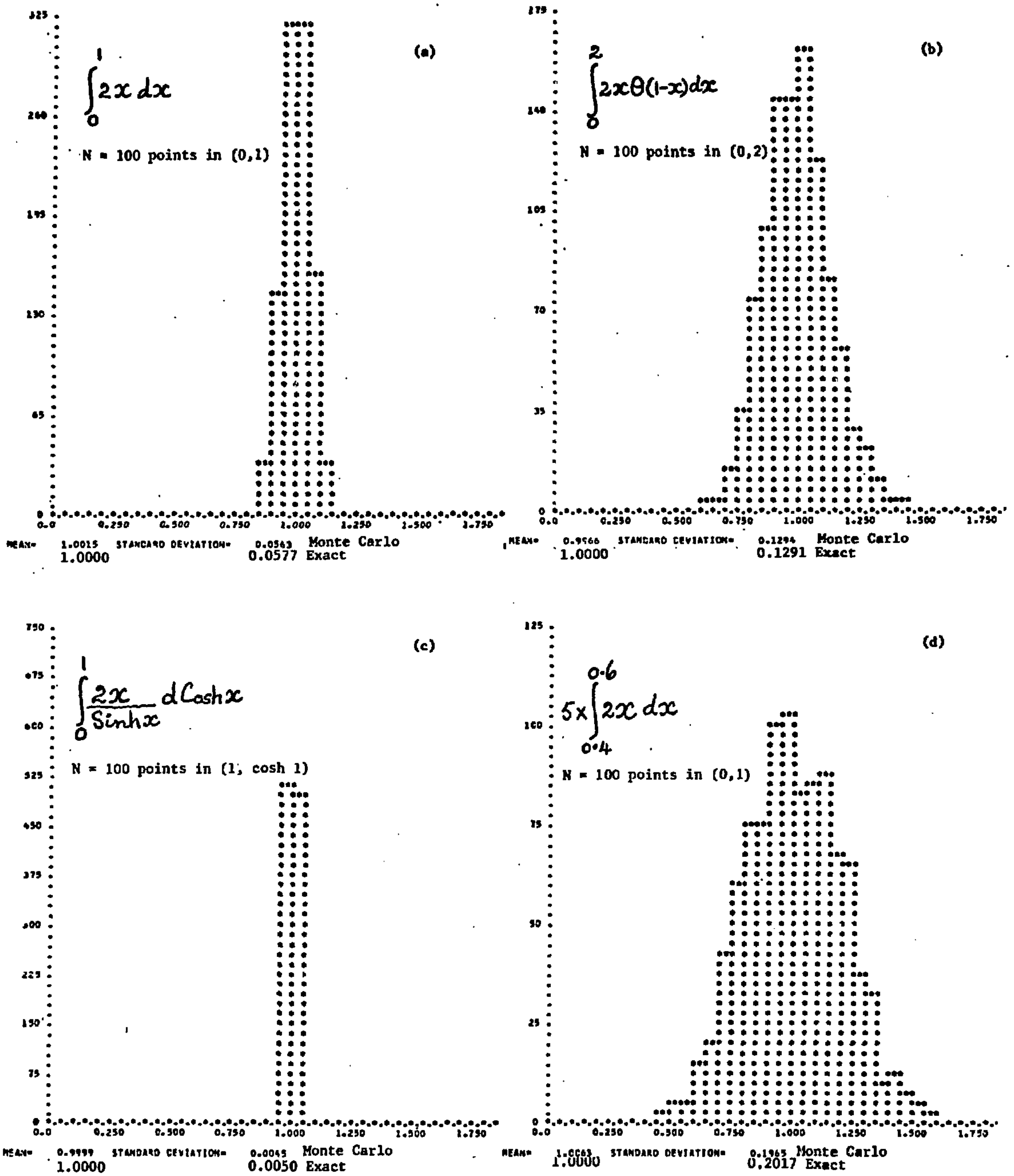

(a)
325
260
195
130
65
0
$\int_0^1 2x\,dx$
N = 100 points in (0,1)
0.0
0.250
0.500
0.750
1.000
1.250
1.500
1.750
MEAN= 1.0015 STANDARD DEVIATION= 0.0563 Monte Carlo
1.0000 0.0577 Exact
(b)
175
140
105
70
35
0
$\int_0^2 2x\theta(1-x)dx$
N = 100 points in (0,2)
0.0
0.250
0.500
0.750
1.000
1.250
1.500
1.750
MEAN= 0.9566 STANDARD DEVIATION= 0.1294 Monte Carlo
1.0000 0.1291 Exact
(c)
750
675
600
525
450
375
300
225
150
75
0
$\int_0^1 \frac{2x}{\text{Sinh}x}\,d\,\text{Cosh}x$
N = 100 points in (1, cosh 1)
0.0
0.250
0.500
0.750
1.000
1.250
1.500
1.750
MEAN= 0.9999 STANDARD DEVIATION= 0.0045 Monte Carlo
1.0000 0.0050 Exact
(d)
125
100
75
50
25
0
$5\times\int_{0.4}^{0.6} 2x\,dx$
N = 100 points in (0,1)
0.0
0.250
0.500
0.750
1.000
1.250
1.500
1.750
MEAN= 1.0063 STANDARD DEVIATION= 0.1965 Monte Carlo
1.0000 0.2017 Exact

Figure 5.

IV. The Programs SIMUL8 and MONTE-PION

(i) General Comments (see figure 6)

The programs are designed to be efficient within the main loop. With minimum options in effect the loop time is about 8 millisec on the UBC IBM system, and rises to 15 millisec with all options in effect. A typical run generates 10,000 (NSTAT) statistics. This number of statistics produces about 1% or better accuracy in the "globally" averaged quantities and about 10% accuracy in small cut quantities. The limits to the electron energy are set by XMIN and XMAX and the gamma ray energy by YMIN and YMAX.* The number of subdivisions within these regions is controlled by NX and NY. If high precision results are required for any subregion it is much better to run the program with say NSTAT = 10,000 statistics all falling within the region of interest, rather than with a massive run of say NSTAT = 1,000,000 falling in a wider region.

The following techniques have been used to increase the speed of the program:

a) Avoiding lengthy algebraic and transcendental operations in the main loop (where possible).

b) Reducing the number of calls to subprograms (where possible).

c) Writing initialisation sections within each subprogram. Where appropriate, all constants are computed on the first entry to the subprogram. A switch, INIT, is then set to prevent subsequent calls entering this region again.

It was found that by using the FORTRAN H compiler only a 10% improvement in speed was achieved. Due to the uncertainties surrounding this compiler its use is not recommended.

(ii) The Co-ordinate system

The programs assume a right handed cartesian co-ordinate system. With the exception of the subroutine BEAMIN any origin and orientation of the co-ordinate system is satisfactory.

Lengths are all specified in centimeters.
Angles are all specified in radian.
Energies, masses and momenta are in MeV, MeV/c^2, MeV/c respectively, or are dimensionless fractions of their maximum values.

In BEAMIN (as it is presently written) the co-ordinate assumptions are:-

i) Z-axis is parallel to the beam direction.
ii) X-axis is the axis of TINA and points into TINA.
iii) Y-axis is vertically up in the laboratory.

*See the Appendix I for an elaboration of relationship between energy scales and pulse heights.

iv) The outside iron face of TINA is the Z=0 plane (see figure 1).

Recommendation: In choosing a co-ordinate system do not necessarily choose the most symmetric or beautiful one, choose one that is tangible and has easy access on the experimental floor!

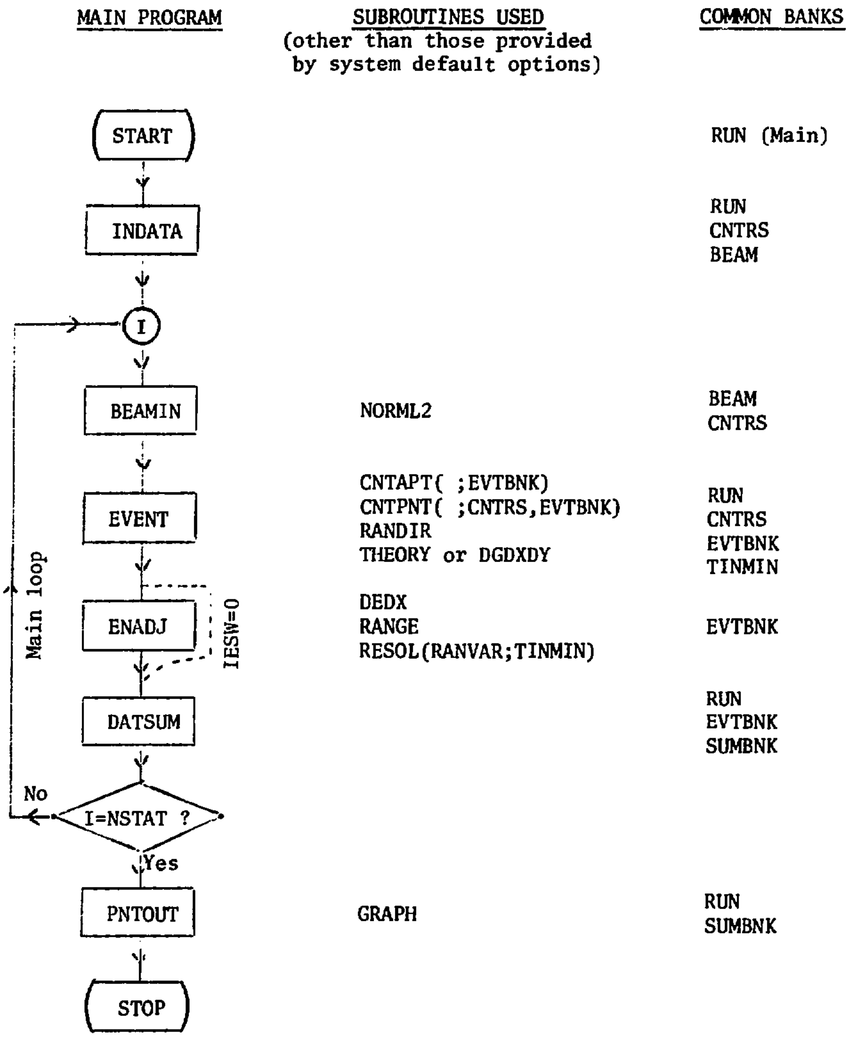


Fig. 4. Structure of the Programs SIMUL8 and MONTE-PION.

iii) Running the program

The source programs are on files SIMUL8•S and MONTE-PION•S, the object programs being on SIMUL8•O and MONTE-PION•O. A typical run from BATCH has a card deck which looks like:

Card 1	Service Request Card	
2	$SIG OPAT T=20 P=20	
3	Password	
4	$RUN SIMUL8•O	
5	R... (run card data)	compulsory
6	B... (beam card data)	compulsory
8	C... counter or aperture data	
9	C...	
:	C...	Any number of cards, (NCNT) in any order, bearing counter or aperture numbers in the range 1 through 30.
:		
:		
:		
6+NCNT	C...	
7+NCNT	$END	
8+NCNT	$SIG	

iv) The Data Cards - their content and format

The data cards are all read by subroutine INDATA which echoes their contents on the line printer in the same format.

The RUN card has the following data string:

Name	NSTAT	XMIN	XMAX	YMIN	YMAX	NX	NY	IESW
eg. R	10000	0.500	1.000	0.300	0.900	10	12	1

FORMAT (1X,I9,4F10.3,3I10)

NSTAT = number of statistics generated

XMIN, XMAX, YMIN, YMAX set lower and upper limits to the variables X,Y.[*]

X,Y refer to electron and gamma ray energies or pulse heights.[*]

NX, NY are the number of bins into which the X and Y distributions which appear in the graphs and tables will be cut. 1 ≤ NX,NY ≤ 20.

IESW is a mode switch.

For IESW = 0 No allowance is made for energy loss or the energy response of TINA or MINA. The energy adjusting subroutine ENADJ is skipped.

= 1 Energy loss and sodium iodide response is allowed for.

= 2 The initial undegraded values of the electron and gamma ray energies are frozen to be equal to XMAX and YMAX and their directions anti-parallel. This option enables the monochromatic energy response to be studied. (Available in SIMUL8 only.)

[*] See the Appendix 1 for a fuller description of the relationship between pulse heights and energies.

The BEAM card has the following data string:

Name	IBMTYP	BM(2)	BM(3)	BM(4)	BM(5)	BM(6)	BM(7)	BM(8)
eg. B	4	-7.620	1.270	5.000	4.445	0.077	10.000	5.000

FORMAT (1X,I9,7F10.3)

The meaning of the items of the data string BM(2)-BM(8) depends on the type of beam (IBMTYP). See the listing of BEAMIN for a description of the beam types available and the meaning of the variables.

The COUNTER cards (or aperture cards) each have the following data string:

Name	IC	CNT(1,IC)	CNT(2,IC)	CNT(3,IC)	CNT(4,IC)	CNT(5,IC)	CNT(6,IC)	CNT(7,IC)	CNT(8,IC)
eg. C	3	-7.420	0.000	0.000	0.881	0.000	0.474	20.320	0.635

FORMAT (1X,I2,F7.3,7F10.3)

The index number IC is the same as that used on the experimental floor with the following restriction, 1 ≤ IC ≤ 30.

For the counter (or aperture) with index number IC the data string has the following meaning:

CNT(1,IC), CNT(2,IC), CNT(3,IC) are the X,Y,Z co-ordinates of the point at the centre of the counter or aperture (on the entry side in the case of a thick counter).

CNT(4,IC), CNT(5,IC), CNT(6,IC) are the X,Y,Z components of the unit vector normal to the plane of the counter. The sense of this unit vector is from the entry to the exit side of the counter. (The sense of this vector is used to test whether a ray penetrated the counter from the expected side.)

CNT(7,IC) is the RADIUS (not diameter) of circular counter or is the side length of a square counter.

CNT(8,IC) is the thickness of the counter or the thickness of material (if any) blocking an aperture

Note: This 8 parameter specification of counters and aperture could in principle be incomplete if:

i) they were all made of different material

ii) they were not circular or square in shape

iii) their orientation (for non-circular) counters required specification.

v) The Named COMMON Banks

There are no unnamed COMMON banks. The contents of the named COMMON will now be described.

```
COMMON/RUN/NSTAT,XMIN,XMAX,YMIN,YMAX,XRANGE,YRANGE,NX,NY,IESW
```

These variables are all on the RUN card except for

XRANGE = XMAX-XMIN

YRANGE = YMAX-YMIN

```
COMMON/BEAM/BM(8)
EQUIVALENCE (IBMTYP,BM(1))
```

These variables are all on the BEAM card.

COMMON/CNTRS/NCNT,CNT(8,30)

These variables are all on the COUNTER or aperture cards except for NCNT, which is the total number of counters and apertures in the system.

COMMON/EVTBNK/P(3),TB,E(3),G(3),X,Y,COSEG,WT(10),KET,KEM,KGT,KGM,THICK

EVTBNK holds the variables for the current event.

P Cartesian coordinates of interaction point.

TB Thickness of material penetrated by beam in stopping counter before coming to rest.

E Unit vector in electron direction.

G Unit vector in gamma ray direction.

X,Y Electron and gamma ray energies in units of the maximum energy, before and after energy adjustment due to slowing and resolution by ENADJ.

COSEG Cosine of the angle between the electron and gamma ray.

WT( ) A string of weights calculated in EVENT and used by DATSUM.

WT(1) Weight for $\mu^+ \to e^+ \nu_e \bar{\nu}_\mu$ (Michel Spectrum).

WT(2) A phase-space-like weight

WT(3) Weight for $\mu^+ \to e^+ \nu_e \bar{\nu}_\mu \gamma$ in SIMUL8 } calculated using
$\pi^+ \to e^+ \nu_e \gamma$ in MONTE-PION } THEORY or DGDXDY

WT(4) ≡ GEOMWT Geometric weight to do with electron back counter solid angle; see CNTPNT.

WT(5) = 0,1 TINA/MINA front aperture function for electrons.

WT(I) $6 \leq I \leq 10$ are free.

KET,KEM = 0,1 Depending on whether CNTPNT put the electron through the TINA or MINA back counter.

KGT,KGM = 0,1 Depending on whether CNTAPT put the gamma ray through the TINA or MINA front and back apertures.

THICK Is the total thickness of material penetrated by the electron allowing for its (slant) trajectory out of the stopping counter and through the various other counters.

COMMON/SUMBNK/A(15),AX(20,15),AY(20,15),AXY(20,20,15),NCOUNT
DOUBLE PRECISION A,AX,AY,AXY

A,AX,AY,AXY are the arrays in which DATSUM assembles the partial sums corresponding to the overall rate, X-distribution, Y-distribution and XY-distribution.

NCOUNT is incremented by unity on each entry to DATSUM. When NCOUNT = NSTAT the arrays are processed to be ready for output by PNTOUT.

V. The Subroutines and Functions

Each subroutine and function is described below, and where it is appropriate any special algebra, geometry or listings are displayed. The subprograms are discussed in alphabetical order for easy access. For full listings consult R. Poutissou of TRIUMF and/or M. Hasinoff of the Physics Department, U.B.C..

BEAMIN

A call to BEAMIN returns the cartesian coordinates P(1), P(2), P(3), of a beam particle stopped in counter #3, (see figure 1), generated in accordance with an assumed beam profile probability distribution. It also finds the thickness, TB, of matter penetrated by the beam in Counter #3. The centre line of the beam is assumed to be parallel to the positive Z-axis. The type of beam profile is determined by a variable IBMTYP which is listed on the Beam card.

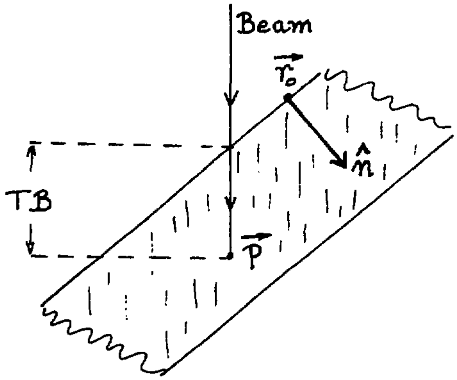
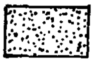


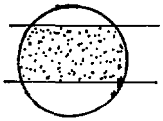
Fig. 7. Beam stopping in counter #3.

The following types of beams exist at the time of writing:

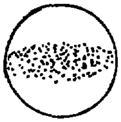
IBMTYP = 1 Parallel beam, uniformly distributed within a rectangular cross-section.



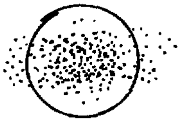
= 2 Parallel beam, uniformly distributed within a vertically cut-off circular aperture.



= 3 Parallel beam, cut-off by a circular aperture with gaussian profiles in the horizontal and vertical directions of differing width.



IBMTYP = 4 Beam multiply scattered by a degrader into a short collimator. Beam profile on the degrader has gaussian profiles in the horizontal and vertical directions of differing width.



As beam 4 is the most complex we describe it further in the form of several notes.

a) Beam card variables

IBMTYP ≡ BM(1) described above = 4

BM(2),BM(3) = (XCENT,YCENT) coordinates of the centre line of the beam.

BM(4) not used.

BM(5) radius of collimator.

BM(6) multiple scattering angle θ_e as used in $\exp\text{-}(\theta/2\theta_e)^2$.

BM(7),BM(8) full width at half height of the beam profiles in the X and Y directions respectively.

b) NORML2 is used to generate both beam position on the degrader (XX,YY) and the multiply scattered beam direction vector (XXN,YYN,1.0).

c) The beam stopping position is chosen uniformly through the counter #3.

d) The following equations are employed to find the intercept of a ray on a plane.

$\vec{r} = \vec{a} + \lambda\hat{\Omega}$ (Equation to ray through $\vec{a}$ with unit direction $\hat{\Omega}$).

$(\vec{r}-\vec{r}_o)\cdot\hat{n} = 0$ (Equation to plane through $\vec{r}_o$ with unit normal $\hat{n}$).

Thus the distance, λ, from $\vec{a}$ to the plane is given by

$\lambda = (\bar{r}_o-\bar{a})\cdot\hat{n}/(\hat{\Omega}\cdot\hat{n})$ and $\vec{r} = \vec{a} + (\bar{r}_o-\vec{a})\cdot\hat{n}\hat{\Omega}/(\hat{\Omega}\cdot\hat{n})$.

CNTAPT(IC,P,U,KOUNT,DX)

This subroutine tests whether a ray starting at point P moving in unit direction U penetrates counter or aperture, IC, from the correct side. If it does, KOUNT=1 and DX=thickness (in cm) of counter penetrated. If it does not, KOUNT=0, and DX=0.0.

CNTPNT

This subroutine "puts an electron through" the TINA or MINA defining counter. More specifically CNTPNT:

a) Decides whether to put the electron through TINA or MINA.

b) Sets the flags KET and KEM accordingly.

c) Finds the point on the defining counters (7 or 8) which is penetrated.

d) Computes the unit electron direction E(1),E(2),E(3).

e) Computes the thickness (cm) of counter penetrated, THICK.

f) Computes the geometric weight GEOMWT $\equiv$ WT(4). The geometric weight is the geometric factor in eq. 39 above, namely $\hat{e}\cdot\hat{n}A/(4\pi r^2)$. The area A is the sum of the TINA and MINA defining counter areas, i.e. $A = \pi R^2{}_{TINA} + \pi R^2{}_{MINA}$. The program regards these defining counters as disjoint patches of a single common area, A.

g) This subroutine (together with several others) needs two orthogonal vectors in a plane normal to a given (unit) vector. The following device is used:
Let (a,b,c) be the given unit vector, $a^2 + b^2 + c^2 = 1$.
Defining $d \equiv (a^2 + b^2)^{\frac{1}{2}}$ we find that (b/d,-a/d,o) and (ac/d,bc/d,-d) are two such unit vectors.

DATSUM

This subroutine at each call summarises the data in a way to be described. On the last call (for which NCOUNT=NSTAT) the data arrays are normalised. DATSUM carries out the following at each call:

a) Combines the computed weights, WT, with the logical variables KET, KGT, KEM, KGM, to produce the weights W relevant to the experiment at hand.

b) It allocates a bin, IX, IY, to the energy variable pair x' and y'. There are NX(NY) bins between XMIN and XMAX (YMIN and YMAX).

c) The double precisioned arrays A(15), AX(20,15), AY(20,15), AXY(20,20,15) have added into them up to 15 different weights, not cut at all, cut into X bins, cut into Y bins,and cut into XY bins respectively.

d) By dividing the theoretical weight into the geometrically modified weights, DATSUM also produces the detection probabilities for each region of x',y' space.

e) The letter A roughly corresponds to the statistic S1.
Note: As written the statistic S2 and its cut versions are not computed by DATSUM, double precision arrays B(15), BX(20,15), BY(20,15), BXY(20,20,15) need adding to the SUMBNK for this purpose.

DEDX(T)

DEDX(T) gives the loss of kinetic energy of the electron per unit path length in a polystyrene counter. The kinetic energy T is in MeV and DEDX is in MeV/cm. The function employed is a piecewise quadratic fit by A.J. (Hannes) Barnard to the electron collision loss tables of M.J. Berger and S.M. Seltzer, National Aeronautics and Space Administration (Washington, D.C.) report NASA SP-3012 (1964). The accuracy of the fit is adequate to the task at hand.

DGDXDY(X,Y)

This function evaluates the differential branching ratio $db/dxdy = (d\Gamma_\gamma/dxdy)/\Gamma$ for the decay $\pi^+ \to e^+ \nu_e \gamma$, and as such it is the main weight function of MONTE-PION. The formula is in accord with:

i) G.I. Opat - Personal Notes 2/April/1977. (J-M. Poutissou and P. Depommier have copies).

ii) P. Scheck and A. Wullschleger, Nuclear Physics B67, 504-517 (1973).

iii) V. Soergel, Proc. SIN Spring School on Weak Interactions and Nuclear Structure, Zuoz. Switzerland, April (1972) Vol. II, P.51.

The inner bremsstrahlung term and the structure dependent terms are evaluated but the interference term is not. This omission has two origins:

a) It is very small under all conditions.

b) The sign of this term is not known (to me at least).

ENADJ

This subroutine adjusts the energy of the positron and gamma ray. For the positron ENADJ:

i) Tests whether the positron has sufficient energy to penetrate all relevant counters. If not WT(5) = 0.0. Uses RANGE(E).

ii) Decreases the electron energy by the collision loss term due to the matter penetrated. Uses DEDX(E).

iii) Adds the annihilation energy $m_e c^2$ to the energy.

iv) Randomly re-allocates the energy in accordance with the NaI detector response function.

Note: The electron energy is taken as the total energy, not as the kinetic energy.

For the gamma-ray ENADJ:

i) Randomly re-allocates the energy in accordance with the NaI detector response function.

The resulting energies are supposed to be analogues of the pulse height put out by TINA and MINA, on a common scale, with X or Y = 1.0 corresponding to an energy EMAX.

EVENT

This is a very major subroutine which works out nearly all the kinematic variables and evaluates the weights for each event. It begins with the stopped particle produced by BEAMIN and puts out its data into the bank EVTBNK.

In order, the following sequence is followed:

i) Electron energy is generated randomly within limits set.

ii) Michel distribution weight is evaluated and put in WT(1).

Note: The philosophy of calculating a "singles" event in parallel with the "doubles" events is very sound, as these are also collected by the experiment. The "singles" data forms a basic check on the experiment.

iii) Gamma ray energy is evaluated randomly within limits set.

iv) Cosine of the angle between the electron and gamma ray is assigned.

a) In SIMUL8 ($\mu^+ \to e^+ \nu\bar{\nu}\gamma$) this is done randomly in accordance with the limits set by eq. 31c.

b) In MONTE-PION ($\pi^+ \to e^+ \nu\gamma$) this angle is directly computed from the energies.

v) In SIMUL8, THEORY evaluates the differential branching rate for $\mu^+ \to e^+ \nu\bar{\nu}\gamma$. In MONTE-PION, DGDXDY evaluates the differential branching rate for $\pi^+ \to e^+ \nu\gamma$.

vi) The electron (which is common to $\mu^+ \to e^+ \nu\bar{\nu}$ and $\mu^+ \to e^+ \nu\bar{\nu}\gamma$ or to $\mu^+ \to e^+ \nu\bar{\nu}$ and $\pi^+ \to e^+ \nu\gamma$) is put through the defining counter (#7 and #8) by CNTPNT.

vii) Aperture acceptance tests are performed on the electron.

viii) The gamma ray direction is generated.

ix) Aperture acceptance tests are performed on the gamma ray.

Note 1 WT(1) is theoretical singles weight.
WT(2) is the theoretical phase space (only) doubles weight.
WT(3) is the theoretical doubles weight.
WT(4) assigned by CNTPNT is the geometric weight.
WT(5) is the electron front aperture acceptance weight. It is changed to zero by ENADJ if the electron fails to penetrate the required amount of matter.

Note 2 In SIMUL8 (only) if the RUN card switch, IESW=2, the electron energy is set to XMAX, and the gamma energy to YMAX. The angle between the electron and gamma ray is set to 180°. Also WT(1) = W(3) = 1.0. This feature was introduced to assist with experimental diagnostics.

GRAPH(A,AM,N)

Plots the graph of N values of a non-negative function stored in the array A. *The maximum value of the string in A is AM, which must satisfy* $1.0 < AM < 10.0$. *GRAPH also* selects a pleasing height for the graph.

INDATA

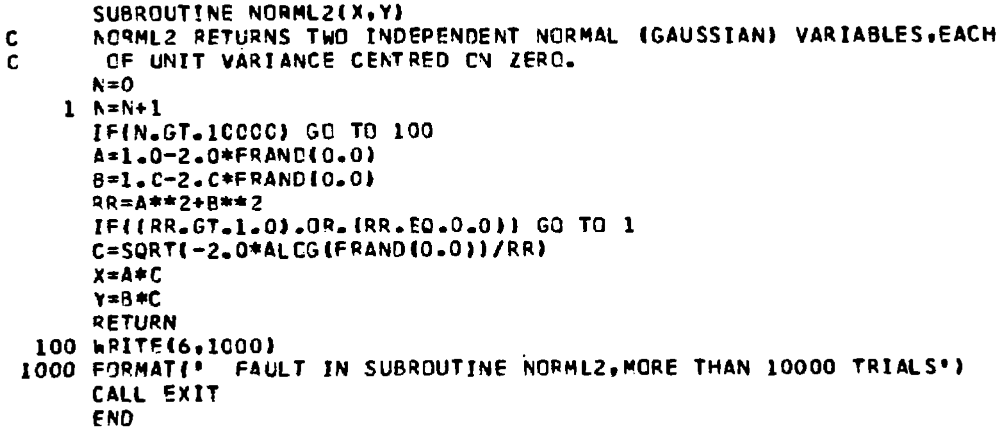
Reads in data cards in formats already discussed.

MAIN

See figure 6. If IESW = 0, ENADJ is skipped.

NORML2(X,Y)

Returns two independent variables X,Y each of which is normally distributed about zero with unit variance. The algorithm on which it is based is exact. In spite of the use of transcendental operations, it was found that NORML2 was faster (by a factor of 2) than two calls to the usual generator* of unit normal variates based on the central limit theorem.



```
      SUBROUTINE NORML2(X,Y)
C     NORML2 RETURNS TWO INDEPENDENT NORMAL (GAUSSIAN) VARIABLES,EACH
C      OF UNIT VARIANCE CENTRED ON ZERO.
      N=0
    1 N=N+1
      IF(N.GT.10000) GO TO 100
      A=1.0-2.0*FRAND(0.0)
      B=1.0-2.0*FRAND(0.0)
      RR=A**2+B**2
      IF((RR.GT.1.0).OR.(RR.EQ.0.0)) GO TO 1
      C=SQRT(-2.0*ALOG(FRAND(0.0))/RR)
      X=A*C
      Y=B*C
      RETURN
  100 WRITE(6,1000)
 1000 FORMAT('  FAULT IN SUBROUTINE NORML2,MORE THAN 10000 TRIALS')
      CALL EXIT
      END
```

*Such a generator function is:

```
   FUNCTION GAUSS (0.0)
   GAUSS = -6.0
   DO 1 I=1,12
 1 GAUSS = GAUSS+FRAND (0.0)
   RETURN
   END
```

PNTOUT

Organizes prints and graphs the contents of the SUMBNK. As presently written it is adequate for the programs SIMUL8 and MONTE-PION. By way of criticism it needs the following modifications:

i) Print and graph control from a run card.

ii) Better graphics, including possibly a contour plot of AXY.

iii) Provision for the "error" statistic output, $\sqrt{S2}$, if this is ever included in DATSUM.

iv) Generally greater flexibility.

RANDIR(CS,SN)

Finds the sine and cosine of a direction in the plane, without finding the angle itself. RANDIR uses no transcendental functions at all and hence is very fast.

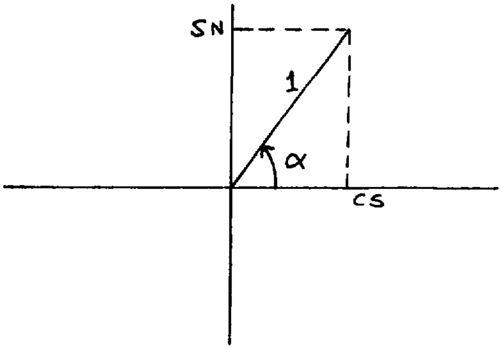


Fig. 8. The angle α is uniform in (0,2π).

```
      SUBROUTINE RANDIR(CS,SN)
C**** RANDIR FINDS THE SINE AND COSINE OF A RANDOM ANGLE BY THE METHOD
C      OF VON NEUMANN.
    1 A=FRAND(0.0)
      B=1.0-2.0*FRAND(0.0)
      A2=A**2
      B2=B**2
      C2=A2+B2
      IF(C2.GT.1.0) GO TO 1
      CS=(A2-B2)/C2
      SN=2.0*A*B/C2
      RETURN
      END
```

RANGE(T)

Gives the range (cm) of an electron of kinetic energy T(MeV) in polystyrene. The function RANGE was constructed in a similar manner to DEDX and from the same reference; see DEDX.

RANVAR(XMIN,XMAX,N,P)

This function implements a general procedure for generating a random variable for an arbitrary one dimensional probability distribution.

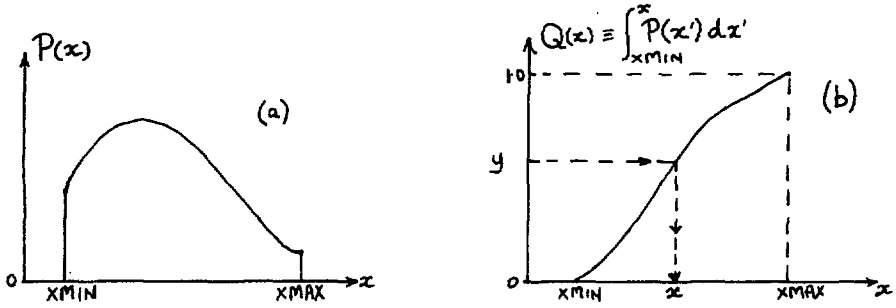


Fig. 9. A probability distribution function P(x) is shown in (a) and its integral Q(x) in (b).

Let dp be the probability of finding x in (x,x+dx). Then

$$dp = P(x)dx \tag{45}$$

defines a probability distribution function. Let Q(x) be defined by

$$Q(x) = \int_{XMIN}^{X} P(x)dx. \tag{46}$$

It is easy to show that if y is uniformly distributed in (0,1) then x given by

$$x = Q^{-1}(y) \tag{47}$$

has the required distribution.

RANVAR expects P(x) to be specified at N equally spaced values of X in the interval (XMIN,XMAX). The consecutive values of P(x) are given in the array P, the 1st being P(XMIN) and the last P(XMAX).

On an initial call RANVAR integrates P(x) by passing a cubic polynomial through each set of 4 consecutive points. The end intervals are treated separately. The values of the integrals at the N points are stored in the array PI, and the differences of the integrals at successive points in the array DPI.

On every call RANVAR "looks up" the value of the required random variable by linear interpolation between the points.

Note: i) RANVAR could be made a little faster by using a "binary search" for the required random variable

ii) RANVAR, as written, can only be initialised once, thus subsequent calls always are evaluated using the first probability distribution function.

```
      FUNCTION RANVAR(XMIN,XMAX,N,P)
C**** RANVAR RETURNS A RANDOM VARIABLE IN THE RANGE (XMIN,XMAX) BASED ON
C      A PROBABILITY DISTRIBUTION GIVEN AT N EQUALLY SPACED POINTS.THE
C      FIRST POINT CORRESPONDS TO XMIN,THE LAST TO XMAX.THE PROBABILITY
C      DISTRIBUTION,WHICH NEED NOT BE NORMALISED,IS TAKEN AS A CUBICALLY
C      SMOOTHED FUNCTION THROUGH THE POINTS.
      DIMENSION P(N),PI(1000),DPI(1000)
      DATA INIT/0/
      IF(INIT.NE.0) GO TO 10
      INIT=1
      DPI(2)=9.0*P(1)+19.0*P(2)-5.0*P(3)+P(4)
      N1=N-1
      DO 1 J=3,N1
    1 DPI(J)=-P(J-2)+13.0*(P(J-1)+P(J))-P(J+1)
      DPI(N)=P(N-3)-5.0*P(N-2)+19.0*P(N-1)+9.0*P(N)
      PI(1)=0.0
      DO 2 J=2,N
    2 PI(J)=PI(J-1)+DPI(J)
      DO 3 J=2,N
      DPI(J)=DPI(J)/PI(N)
    3 PI(J)=PI(J)/PI(N)
      DX=(XMAX-XMIN)/FLOAT(N-1)
   10 R=FRAND(0.0)
      DO 11 J=2,N
      T=R-PI(J)
      IF(T.LE.0.0) GO TO 12
   11 CONTINUE
   12 RANVAR=XMIN+DX*(FLOAT(J-1)+T/DPI(J))
      RETURN
      END
```

RESOL(E,IEORG,ITORM)

The function RESOL allocates to the deposited energy E (MeV) a random variable, the pulse height RESOL (MeV), according to the pulse height response (probability) function. (See Appendix I for details of the pulse height scale.)

RESOL is based on 3 hypotheses concerning the response function. If E is the deposited energy and H is the corresponding pulse height these hypotheses are:-

(a) $H_0 \propto E$, where H_0 is the most probable pulse height. (In RESOL, $H_0 = E$).

(b) $\Delta H \propto E^{\frac{1}{2}}$, where ΔH is the full width at half height of the response function.

(c) The response function profile is universal.

RESOL implements these hypotheses by the use of a data set stored by a BLOCK DATA statement. The data set includes:

i) The four fractional widths, $\Delta H/H_0 \equiv F$, for the cases electrons and gamma rays into TINA and MINA.

ii) The energy EZERO at which these four fractional widths pertain.

iii) The universal response function in standard form. The N (unnormalised) values of this function are held in RESFUN. The standard form requires that the most probable point have abscissa zero, and that the full width of half height be unity. The lower and upper values at which the response function goes to zero are RL and RU, (RL <0.0). See figure 10.

RANVAR implements the generation of a random variable in accordance with the universal response function.

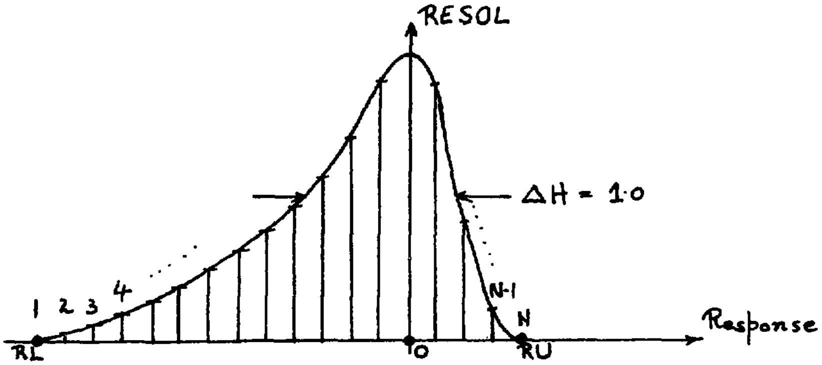


Fig. 10. The data representing the universal response function in standard form.

```
      FUNCTION RESUL(E,IEORG,ITORM)
C**** RESUL SMEARS THE ENERGY OF ELECTRONS AND GAMMA RAYS IN TINA AND
C       MINA ACCORDING TO THE EXPERIMENTAL PROFILE.
      COMMON /TINMIN/RL,RU,N,RESFUN(100),F(2,2),EZERO,EMAX,DEMIN
C       IEORG=1,2 FOR ELECTRON, GAMMA-RAY
C       ITORM=1,2 FOR TINA,MINA
C      E AND RESUL ARE IN MEV
      Z=E/EZERO
      Z=Z+SQRT(Z)*F(IEORG,ITORM)*RANVAR(RL,RU,N,RESFUN)
      RESUL=EZERO*Z
      RETURN
      END

      BLOCK DATA
      COMMON /TINMIN/RL,RU,N,RESFUN(100),F(2,2),EZERO,EMAX,DEMIN,EANNIH
      DATA RL,RU,N,EZERO,EMAX/-4.7319,0.9464,19,52.8297,52.8297/
C     PRESENT DATA IS FROM D.BERGHOFER FROM PION CEX DATA (26/3/1977).
      DATA RESFUN/0.0,8.6,21.1,33.6,44.8,67.1,92.1,117.2,142.2,
     1 238.9,406.9,559.7,774.9,1187.3,1918.8,2472.2,1504.0,374.5,
     *0.0,81*0.0/
      DATA EANNIH/0.5110034/
C
C     THE N VALUES OF RESFUN SPECIFY NUMERICALLY THE UNIVERSAL SODIUM IODIDE
C     RESPONSE FUNCTION ON A SCALE IN WHICH THE PEAK IS AT 0.0 AND THE FULL W
C     WIDTH AT HALF HEIGHT IS 1.0
C     RL, RU ARE THE LOWER AND UPPER VALUES AT WHICH THE RESPONSE GOES TO 0.0
C     ON THIS SCALE .
      END
```

THEORY(X,Y,DL)

Evaluates the differential branching ratio for the decay $\mu^+ \to e^+ \nu\bar{\nu}\gamma$ and as such is the main weight function of SIMUL8. The formula is taken from G. Källén, Springer Tracts in Modern Physics 46, 96 (1968). The electron mass has been set to DL = $1-\text{Cos}\theta_{e\gamma}$.

VI. References

1. J.M. Hammersley and D.C. Handscomb. Monte Carlo Methods (Halsted Press (Wiley) U.S.A., or Methuen U.K., 1965).

2. J.H. Halton. A retrospective and prospective survey of the Monte Carlo Method. S.I.A.M. Rev. 12, 1-63, (1970). (This review is particularly valuable for its long list of references, some of which are to physics applications.)

3. J.P.C. Kleijnen. Statistical Techniques in Simulation, Vols. I & II (Marcel Dekker, N.Y., 1974-5).

4. F.E. James. The CERN program FOWL and its long writeup. (CERN Program Library - Geneva, Switzerland) (FOWL evaluates integrals of arbitrary matrix elements over the covariant phase space of any number of particles using MONTE-CARLO methods.)

APPENDIX 1

Important Note on Energies and Pulse Heights

Before entering ENADJ, X and Y are the total energies of the electron and gamma ray in units of the maximum theoretically allowed energy, EMAX.
(EMAX = $M_\mu/2$ in SIMUL8). Thus $0.0 \leq X,Y \leq 1.0$.

In ENADJ, X and Y are converted to an MeV energy scale. The (positive) electron energy, ENE, is adjusted to allow for collision loss in the counters and the annihilation energy, whereas the gamma ray energy ENG is not adjusted at all. At this stage ENE and ENG are the energies dumped into the Sodium Iodide crystals by the electron and gamma ray.

ENADJ now sends ENE and ENG through the crystal response routine RESOL. The output variables of RESOL, are also called ENE and ENG but now represent pulse heights in the standard convention.

The standard convention for pulse heights is described as follows:

i) Zero pulse height corresponds to zero energy dumped into the crystal.

ii) The most probable pulse height when energy E is deposited is assumed to be proportional to E on physical grounds. For this reason, in the standard convention for pulse heights the most probable pulse height for energy E MeV deposited is taken to be numerically equal to E. We note that the pulse heights corresponding to deposited energy E can be less than or (slightly) greater than E.

Finally, ENADJ divides the pulse heights ENE and ENG by EMAX to produce the dimensionless variables, again called X and Y. We note that X and Y are positive variables whose range can extend a little over unity.

When IESW=0, ENADJ is not involved and X,Y remain energy variables.

The run card variables XMIN, XMAX, YMIN, YMAX set limits on the ranges of the output variables X and Y.

If IESW = 0 these parameters set the limits to electron and gamma ray energies.

If IESW ≠ 0 these parameters set the limits to the electron and gamma ray pulse heights. In this case the program has to compute the corresponding limits to to the ranges of the energies. This is done in an initialisation section of EVENT and the limits are denoted XL, XU, YL, YU. These limits are chosen to be just wide enough, so that no energy point excluded could have produced a pulse height in the desired range.