SAGE

A General System for Monte Carlo Event Generation with Preferred Phase Space Density Distributions

by

Jerry Friedman

University of California
Lawrence Radiation Laboratory

Group A Programming Note
No. P-189 (revised)
January 1971

and

Stanford Linear Accelerator
Center
Computation Group Technical
Memo No. 145
December 1972

TABLE OF CONTENTS

I. Introduction

SAGE is a set of FORTRAN subroutines that generate Monte Carlo events in Lorentz invariant phase space. These events may be used to simulate particle physics experiments or evaluate phase space integrals.

In addition to generating events that correspond to a uniform population of phase space, SAGE can generate events with other phase space density distributions. This will increase generation efficiency if the generated phase space density distribution of the Monte Carlo events approximates the spin averaged matix element squared of the user's model better than a constant. The additional phase space density distributions available in SAGE include Breit-Wigner[1] invariant mass distributions, e^{at} and $1/(t-a^2)^2$ four momentum transfer squared distributions and general decay angular distributions. SAGE has standard options for generating events that correspond to 1) a uniform density in Lorentz invariant phase space, 2) limited transverse momenta of final state particles (peripheral phase-space), 3) limited successive four momentum transfers squared in multiperipheral chains (multiperipheral phase space). In addition to these standard options, SAGE can be used to generate Monte Carlo events with a wide variety of different phase space density distributions. These distributions can be adjusted to the user's particular problem in order to achieve the best possible efficiency.

SAGE can generate the four vectors of the final state particles in any Lorentz frame specified by the user. It can be used to calculate total and differential cross sections predicted by dynamical models. It can simulate experiments with any distribution of initial state center of mass energy, and it can be used to generate events according to several different frequency distributions for the same problem.

The input to SAGE consists of the kinematic parameters of the initial state (center of mass energy or beam momentum, beam and target mass), specification of the Lorentz frame in which the final state particles' four-vectors are to be expressed, and the parameters that characterize the phase space density distribution of the generated events. The output of SAGE consists primarily of a set of four vectors for the final state particles in the Lorentz frame specified by the user. The other final state kinematic quantities that are also available depend upon the SAGE options employed.

The user can use this output to calculate values for matrix elements predicted by dynamical models for the purpose of simulating experiments or calculating phase space integrals. SAGE may be used in conjuction with data summarizing programs (for example SUMX [1] or KIOWA [2] in order to histogram, scatter plot, or otherwise summarize the Monte Carlo data. The output of SAGE can also be used as input to data reduction programs for the study of experimental biases.

II. Initialization

Before any events are generated or whenever the initial state parameters are changed SAGE must be initialized. This may be done with one of the following FORTRAN statements:

1) CALL INITL (BEAM,TARGET, ECM, PLAB,P)

2) CALL INITL (BEAM, TARGET, ECM, 3HEND)

3) CALL INITLϕ(ECM)

BEAM= mass of beam particle

TARGET= mass of target particle

ECM = initial (and final) state center of mass energy

PLAB = Laboratory beam momentum

P(4) = four vector of the initial state. This defines the Lorentz frame

in which the final state particles' four vectors are to be generated.

For the last two calls all of the parameters are input. For the first one all are input except the third, ECM, which is output returned to the user.

As mentioned above the fifth argument in the first call, P, defines the Lorentz frame in which the final state particles' four vectors are to be generated. For example, P = (0,0,0,ECM) will cause them to be generated in the center of mass system. P = (0,0,PLAB,ELAB + TARGET) will cause them to be generated in the laboratory system (the Lorentz frame where the target is at rest). However, if the final state results from the decay of a particle with four momentum P, or if the beam and target are not collinear (as in a colliding beam experiment) then the components of P may have different values.

Since the center of mass and laboratory systems are the ones most frequently used, special flags may be used to indicate them. P= 2HCM or P = 3HEND indicates the center of mass system and P = 3HLAB indicates the laboratory system. The fourth argument in the calling sequence, PLAB, is only used when P is set to these special flags. If P is set to an arbitrary four vector then INITL ignores the PLAB entry. However, in all cases ECM is computed and returned to the user.

The second call assumes P = 2HCM and that ECM is input supplied by the user. The third call may only be used if there is no four-momentum transfer squared dependence in the final state density distribution. In this case, ECM is also input supplied by the user.

COMMON / INIT / BM(4),TG(4)

BM is the four vector of the beam and TG that of the target, expressed in the same Lorentz frame as the final state particles. This common storage is filled by SAGE in the call to SUBROUTINE INITL (but not INITLØ).

III. Event Generation

A Monte Carlo event is obtained from SAGE by one or more calls to SAGE subroutines. The number of calls for each event, and the subroutines called are goverened by the phase space density distribution desired. SAGE has three subroutines that can generate an entire event with one call. These routines generate events in phase space corresponding to uniform density distribution (GOGEN), limited transverse momenta of the final state particles (GENIS), and limited successive four momentum transfers squared in multi-peripheral chains (GODEL). More general phase space density distributions are possible. These require more than one subroutine call to SAGE for each generated event. Each of these options are discussed in turn below.

A. GOGEN (Uniform Phase Space)

GOGEN employs the technique of Ref. 3 to generate final state particle four vectors corresponding to a uniform phase space density distribution. GOGEN may be used to generate all or some of the final state particles. GOGEN generates a multiparticle system $A=(A_1, A_2, \ldots, A_N)$, by factoring it into a special set of two body decay vertices. The first vertex recoils $A_1 \ldots A_{N-1}$ from A_N. The second recoils $A_1 \ldots A_{N-2}$ from A_{N-1} and so on until the last recoils A_1 from A_2. A_1 may be a multiparticle system but $A_2 \ldots A_N$ must be single particles.

This multiparticle system is generated by the single subroutine call

CALL GOGEN (U,P,N,S,U1,P1)

1. Input Quantities

U = invariant mass of A system

P = four vector of A system

P(1) = 2HCM if A is expressin the center of mass
= 3HLAB if A is expressed in the laboratory

N(1) = number of particles comprising A system

N(2) = number of particles comprising A_1 system

S = rest masses of $A_1 \cdots A_N$

S(1) = sum of rest masses of A_1 system

S(2) = rest mass of A_2

.
.
.

S(N) = rest mass of A_N

Note that if A represents the whole final state then U must equal ECM and P must be the same as entered (or implied) in the fifth argument in the call to INITL.

2. Output Quantities

U1 = invariant masses of multiparticle recoiling systems

U1(1)= invariant mass of A_1 system

U1(2)= invariant mass of $A_1 + A_2$ system

U1(3)= invariant mass of $A_1 + A_2 + A_3$ system

.
.
.

U1(N-1)= invariant mass of $A_1 + A_2 + A_3 \ldots + A_{N-1}$ system

P1 = four vectors of $A_1 \cdots A_N$ in the specified Lorentz frame

P1(4,1)=four vector of A_1

P1(4,2)=four vector of A_2

.
.
.

P1(4,N)= four vector of A_N

If only the invariant masses U1(1) ... U1(N-1) are required the user can suppress the four vector generation in GOGEN by setting P1(1) = 3HEND. For this case the second argument in the calling sequence P is ignored.

B. GENIS (Peripheral Phase Space)

GENIS employs the technique of Ref. 4 to generate final state particle four fectors with a phase space density corresponding to $exp\left(-\sum_{i=1}^{N} p_i^2 /2r\right)$ and is discussed in detail there. The p_i are the components of the particles' momentum transverse to the beam direction, and r is a parameter that characterizes the degree to which these components are limited.in the generation. N is the number of particles to be generated. As with GOGEN, GENIS may be used to generate all or part of the final state. Events of this type are obtained with the single subroutine call:

CALL GENIS (U,P,N,S,P1,R)

1. Input Quantities

U = invariant mass of all N-particles.

P = four vector of all N particles

P(1)= 2HCM if in center of mass system[2]

N = number of particles to be generated

S = rest masses of the N particles

S(1)= rest mass of first particle

.

.

S(N)= rest mass of N th particle

R = parameter that characterizes the limitation of transverse momenta and represents $(<p_i^2>)^{\frac{1}{2}}$ for the generated particles.

Note that if the N generated particles represent the entire final state then U must equal the center of mass energy and P must be the same as entered (or implied) in the fifth argument in the call to INITL.[2]

The parameter, R, determines the distribution of the transverse momenta of the generated particles. It is related to the standard deviation of the Gaussian distribution by $R= (<p_i^2>)^{\frac{1}{2}} = 2r^2(N-1)/N$. The value of R should

be adjusted to obtain the best generation efficiency for the users particular matrix element squared. Experimentally it is found that $R \simeq .4$ GeV, so that this is probably a good starting value. Procedures for adjusting generation parameters in order to maximize generating efficiencies in SAGE is discussed in section VIII.

2. Output Quantities

Pl = four vectors of the N generated particles in the specified Lorentz frame.

P(4,1)= four vector of first particle

.
.
.

P(4,N)= four vector of N th particle

C. GODEL (Multiperipheral Phase Space)

GODEL employs the technique of Ref. 5 to generate final state particle four vectors with a phase space density corresponding to

$$\exp\left(\sum_{i=1}^{N-1} a_i t_i\right) \qquad (1)$$

or

$$\prod_{i=1}^{N-1} \frac{1}{(t_i - a_i^2)^2} \qquad (2)$$

and is discussed in detail there. The t_i are the successive four momentum transfers squared in a multiperipheral chain. The parameters, a_i, characterize the distribution of the t_i for the generated particles. As with GOGEN and GENIS, GODEL may be used to generate all or part of the final state. GODEL factors the N particle system into two body decay vertices in exactly the same manner as GOGEN. The notation used below is the same as in section III-A.

Events with phase space densities given by Eqns (1) or (2) are obtained by the subroutine call

CALL GODEL (U,P,N,S,Pl,PARM,FLAG)

1. Input Quantities

U. = invariant mass of A system

P = four vector of A system

P(1)= 2HCM if A is expressed in the center of mass

= 3HLAB if A is expressed in the laboratory

N(1)= number of particles comprising A system

N(2) = number of particles comprising A_1 system

S = rest masses of $A_1 \cdots A_N$

S(1) = sum of rest masses of A_1 system

S(2) = rest mass of A_2

.
.
.

S(N) = rest mass of A_N

PARM = the four parameters that characterize the phase space density distribution (See Ref. 5)

PARM(1) = b_o Ref. 5 - Eqn. 20

PARM(2) = E_o Ref. 5 - Eqn. 20

PARM(3) = d_1 Ref. 5 - Eqn. 25

PARM(4) = d' Ref. 5 - Eqn. 25

FLAG = indicator of type of phase space density distribution (Eqn. (1) or (2)) and defines the t's used in eqn's (1) and (2).

= 4HBDEL: Eqn. (1), t defined with respect to beam

= 4HTDEL: Eqn. (1), t defined with respect to target

= 4HBPOL: Eqn. (2), t defined with respect to beam

= 4HTPOL: Eqn. (2), t defined with respect to target

Note that if A represents the whole final state then U must equal ECM and P must be the same as entered (or implied) in the fifth argument in the call to INITL.

As indicated above, the four parameters, PARM, determine the phase space density distribution of the generated events. As discussed in Ref. 5, the values of these parameters should be adjusted to obtain the best generation efficiency for the user's particular matrix element squared. Table 2 of Ref. 5 can provide useful starting points. Procedures for adjusting generation parameters in order to maximize generating efficiency in SAGE is discussed in Section VIII.

2. Output Quantities

Pl = four vectors of $A_1 \ldots A_N$ in the specified Lorentz frame

Pl(4,1)= four vector of A_1

Pl(4,2)= four vector of A_2

.
.
.

Pl(4,N)= four vector of A_N

D. Generalized Phase Space Density Distributions

SAGE generates an n - particle final state by factoring it into (n-1) two body systems and generating each system as a two body decay, $A \rightarrow A_1 + A_2$[3]. Here A is a parent system comprised of $N \leq n$ particles and A_1 and A_2 are its daughters, each comprised of M_1 and M_2 particles respectively. Note that $N \geq 2$, M_1 and $M_2 \geq 1$, and $N = M_1 + M_2$. The way in which the final state is factored determines which invariant masses and four momentum transfers squared can be given Breit-Wigner or e^{at} or $1/(t-a^2)^2$ density distributions. A particular factoring of an n - particle final state yields a particular set of n - 2 invariant masses and n - 1 momentum transfers squared that may (but need not) be given these density distributions. The user specifies the phase space density distribution of the generated events both by the way in which he factors the final state into two body decays and the way each individual two body decay is generated. The choice as

to how to factor the final state is decided by the user by coding a SAGE "event type." The event type consists of a series of calls to SAGE subroutines. Each call generates a two-body decay vertex. The user connects these vertices to form the final state [3].

Consider a decay vertex $A \rightarrow A_1 + A_2$. The invariant masses of the A_1 and A_2 systems may be generated according to phase space or Breit-Wigner density.(1) The decay angular distribution of A_1 (or A_2) in the A rest frame may be generated flat or according to e^{at} or $1/(t-a^2)^2$, where t is the four momentum transfer squared from either the beam or target. This decay angular distribution may also be generated according to an arbitrary angular distribution supplied by the user, with either the beam, target, or helicity directions as the z - axis.

In order to generate this vertex the user must make <u>two</u> calls to SAGE :

```
CALL RY(S1,S2,U,P,E,D)
CALL GO(P1,P2,U1,U2)
```

The first call provides the necessary input to generate the vertex and the second returns the output from the generation.

1. <u>Input Quantities</u>

S1(1) = sum of rest masses of A1 system

S1(2) = number of particles comprising A1 system

S2(1) = sum of rest masses of A2 system

S2(2) = number of particles comprising A2 system

U = invariant mass of A system

P(4) = four vector of A

P(1) = 2HCM if A is expressed in the overall center of mass

= 3HLAB if A is expressed in the laboratory

E(1) = mass of Breit-Wigner for generating A_1 invariant mass

= 0.0 for phase space generation of A_1 invariant mass

E(2) = width of Breit-Wigner for A_1 mass

E(3) = mass of Breit-Wigner for generating A_2 invariant mass
= 0.0 for phase space generation of A_2 invariant mass

E(4) = width of Breit-Wigner for A_2 mass

D(1) = $\phi.\phi$ for flat decay angular distribution of A_1 and A_2 in A - rest frame
= 4HBDEL for e^{at} generation with respect to the beam direction
= 4HTDEL for e^{at} generation with respect to the target direction
= 4HBCOS for I(cos θ) generation with respect to beam direction
= 4HTCOS for I(cos θ) generation with respect to target direction
= 4HHCOS for I(cos θ) generation with respect to helicity direction
= 4HBPOL for $1/(t-a^2)^2$ generation with respect to beam direction
= 4HTPOL for $1/(t-a^2)^2$ generation with respect to target direction

D(2) = $1.\phi$ for generating A_1 as above
= $2.\phi$ for generating A_2 as above

D(3) = a, for e^{at} or $1/(t-a^2)^2$ generation
= cosθ, for I(cos θ) generation

Note that for the first call for an event P must be the same as the fifth argument in the call to INITL (or implied if the calling sequence is truncated)

2. Output Quantities

P1(4,1) = four vector of A_1 in specified Lorentz frame

P1(4,2) = four vector of A_1 in A rest frame

P2(4,1) = four vector of A_2 in specified Lorentz frame

P2(4,2) = four vector of A_2 in A rest frame

U1 = invariant mass of A_1 system

U2 = invariant mass of A_2 system

COMMON / MDEL / T,COSINE

Whenever a vertex is generated according to an e^{at} or $1/(t-a^2)^2$ distribution, upon return form GO, T will contain the t generated for that vertex and COSINE will contain the corresponding cosine.

3. Special Calls

The calling sequences of RY and GO may be shortened (optionally) for certain standard situations. Let X= 3HEND, then:

CALL RY(S1,S2,U,P,E,X) implies D(I) = ϕ, I = 1, 3

CALL RY(S1,S2,U,P,X) implies E(I) = ϕ, I = 1,4 also.

CALL RY(S1,S2,U,X) implies P = 2HCM also

CALL Rϕ (S1,S2,U) same as CALL RY(S1,S2,U,X)

CALL GO(P1,P2,U1,X) for S2(2) = 1.ϕ

CALL GO(P1,P2,X) for S1(2) = S2(2) = 1.ϕ

In order to generate the invariant mass(s) U1(and U2) at a vertex, the generator needs only S1, S2, and U. Thus, the N-2 invariant masses (for an N-particle final state) may be generated without generating any four vectors. Alternately, if only four vectors above a certain vertex are needed then those below the vertex need not be generated. (However, the masses below that vertex must still be generated whether used or not.) Since most of the computational time in generating an event goes in constructing four-vectors, considerable computation time can be saved by only generating masses when the four vectors are not needed. Note that if a vertex decay angular distribution is to be generated other than flat, all of the four vectors above and includ- that vertex must be generated.

In order to generate only masses at a vertex the user must call Gϕ instead of GO with the following calling sequence:

CALL Gϕ(U1,U2) if both A_1 and A_2 are multi-particle systems

or optionally

CALL GØ(U1,X) if A_2 is a single particle

CALL GØ(X) if both A_1 and A_2 are single particles

If the vertex is to be generated flat in decay angles, then P may be set to 2HCM in the call to RY (or call RØ) since to generate U1 (and U2), only S1, S2 and U are needed.

COMMON / SWITCH / ISW

ISW is an optional switch that allows the generation of an event in a two step process by making two passes over the same coding.

ISW = 0 (or ignored); no effect

ISW = 1; generate only masses even though calls to RY and GO indicate 4-vectors are to be generated.

ISW = 2; using the masses as input quantities, generate 4-vectors indicated in the calls to RY and GO.

Thus, the user may execute a series of calls to RY and GO with ISW = 1 and generate only masses, then set ISW = 2 and re-execute the same calls to generate the four vectors using the masses generated on the ISW = 1 series of calls.

For this mode of operation, it is necessary to store all of the masses returned on the ISW = 1 series of calls in separate variable locations local to the calling routine. These will then be used to generate the 4-vectors on the ISW = 2 calls.

The user may also generate the masses in an arbitrary way independent of SAGE and by setting ISW = 2, have SAGE generate the four vectors corresponding to these masses. Also, if at any single decay vertex the user wishes to generate an invariant mass independent of SAGE, he may set S1(2) [or S2(2)] eaual to 1.0 even though the invariant mass represents a system of more than one particle. In this case, SAGE will use the value of S1(1) [or S2(1)] as

the input invariant mass for the system.

E. Generating According to a Decay Angular Distribution

As noted above, any vertex may be given an arbitrary decay angular distribution $I(\cos\theta)$. This is accomplished by setting the D-array in the call to RY, for that vertex, as described in section III-D, where $D(3) = \cos\theta$ for each event. This cosine is input supplied by the user. It can be generated as follows:

let $F(\cos\theta) = \int_{-1}^{\cos\theta} I(\cos\theta')\, d(\cos\theta')$

where $I(\cos\theta)$ is the desired decay angular distribution. For each event, generate a random number $0 \leq r \leq F(1)$ and set

$$F(\cos\theta) = r$$

Then $\cos\theta$ for the event is found by solving this equation for $\cos\theta$, ie

$$\cos\theta = F^{-1}(r)$$

IV. Phase Space Weight

The events generated by SAGE are not distributed correctly in Lorentz invariant phase space. A weight must be applied to each event to obtain the correct phase distribution.

After the entire event has been generated, the "phase space" weight for the event is obtained by the subroutine call:

CALL WT(W)

W(1) = "phase space" weight for the event

W(2) = W(1) x phase space density distribution function.

Calling WT also flags the end of an event for SAGE, so that it assumes the next call starts a new event. Thus, if an incomplete event is discarded, WT, INITL, or INITLØ must be called (even though the weight is not used) to initialize SAGE for the next event.

A complete discussion of phase space weighting is given in Ref. 3. Only the details necessary for its use are given here. For section III-D the phase space density distribution function of the generated events is the product of the Breit-Wigner, and e^{at}, $1/(t-a^2)^2$, and $I(\cos\theta)$ distributions applied at each vertex in the event generation. This function represents the density of the generated events in Lorentz invariant phase space.

Let $r(\phi) = \sum_{\text{spins}} |M(\phi)|^2$ be the spin summed Lorentz invariant amplitude squared for a model where ϕ is a point in the phase space. Let $R(v) = \int_{v \leq V} r(\phi)\, d^n\phi$ be the rate into a subvolume of phase space v, that is included in the total phase space volume V, accessible to the system. If the n - particle final state results from the decay of a particle with energy E in some Lorentz frame, then the decay rate (number of decays per particle per time) into the subvolume v in that Lorentz frame is

$$\lambda(v) = 1/\gamma(v) = R(v) / 2E(2\pi)^{3n-4},$$

If the final state results from the collision of two particles then the collision cross section into the subvolume v is

$$\sigma(v) = R(v) / 4m|\vec{p}|(2\pi)^{3n-4}$$

where m is the mass of one of the incident particles and $\vec{p}$ is the momentum of the other particle in the rest frame of the first.

In SAGE the first weight W(1) is normalized so that

$$R(v) = \lim_{N \to \infty} (1/N) \sum_{i=1}^{N} r(\phi_i)\, W_1(\phi_i)\, \theta(v - \phi_i) \qquad (3)$$

The sum is over the generated Monte Carlo events. N is the total number of events generated, $r(\phi_i)$ is $r(\phi)$ evaluated for each event and $W_1(\phi_i)$ is the first weight W(1) returned by SAGE for each event. The step function, $\theta(v - \phi_i)$ is either zero or one depending upon whether ϕ_i is outside or inside the subvolume v. Equation 3 is true independent of the frequency distribution of the generated events. However, for finite N, the statistical accuracy of the

answer is very dependent upon the frequency distribution of the generated events.

The second weight W(2) is normalized so that

$$G(v) = \int_{v \leq V} g(\phi)\ d^n\phi = \lim_{N \to \infty} (1/N) \sum_{i=1}^{N} W_2(\phi_i)\ \theta(v - \phi_i) \quad (4)$$

where $g(\phi)$ is the phase space density distribution function of the generated events.

Thus, to obtain distributions corresponding to $r(\phi)$ each event is weighted with $r(\phi)$ W(1). For distributions corresponding to $g(\phi)$ each event is weighted with W(2). The step function in the above formulae imply that the subvolume of phase space v, is defined by simply discarding those events that are defined to lie outside of it.

V. Variable Center of Mass Energy

In order to more accurately simulate experimental conditions, the center of mass energy supplied to SAGE may vary from event to event. These energies can come from a frequency distribution P(E) determined by the experimental setup if P(E) is proportional to the path length of the experiment as a function of energy. Note that INITL or INITLØ must be called every time the center of mass energy is changed.

VI. More General Phase Space Density Distributions

It is possible to generate Monte Carlo events with densities in Lorentz invariant phase space that are linear combinations of those described above. Let $g_i(\phi)$ be a density distribution available in SAGE. One can generate events for which phase space density is given by

$$g(\phi) = \sum_{i=1}^{M} N_i\ g_i(\phi) / G_i. \quad (5)$$

N_i is the number of events generated with density distribution $g_i(\phi)$ and M is the number of such density distributions. G_i is given $G_i = \int_V g_i(\phi)\ d^n\phi$ and from Eqn. (4) is just the average of W(2) for events generated with

frequency $g_i(\phi)$.

First N_1 events are generated with density $g_1(\phi)$ as prescribed above. Then N_2 events are generated with density $g_2(\phi)$ and so on until N_M events are generated with density $g_M(\phi)$. The total number of events generated is $N = \sum_{i=1}^{M} N_i$. The weight applied to each event in order to obtain distributions corresponding to $r(\phi)$ is $r(\phi)\ W(2) / G_i\ g(\phi)$. Thus the weight for an event depends upon the simple frequency distriubtion $g_i(\phi)$ from which it was generated. For distributions corresponding to $g(\phi)$ each event is weighted with $W(2) / G_i$. Phase space subvolumes are still defined by rejecting events as discussed above.

Note that in order to use this method the values of G_i must be known in advance. These values are most easily obtained by generating a set of events according to each of the simple frequency distributions $g_i(\phi)$ and averaging $W(2)$ for each case. This need only be done once for a given $g_i(\phi)$ and then its value can simply be entered from then on.

VIII. Unweighted Events

SAGE allows the simulation of experiments with transition rate $r(\phi)$ by weighting each event generated by $r(\phi)\ W(1)$ or $R(\phi)\ W(2) / G_i\ g(\phi)$. For some applications, however, unweighted events are required whose density in Lorentz-invariant phase space is given by $r(\phi)$. This can be accomplished by generating along with each event a random number, x, in the interval $(0, x_{max})$, where x_{max} is an upper bound for the event weights. Events are discarded whose weight is less than x, and those not discarded are each given unit weight. These events are distributed randomly with a density $r(\phi)$ in Lorentz-invariant phase space. The upper bound, x_{max}, need not be the least upper bound for the event weights, but the closer it is to the least upper bound the more efficient the method becomes. However, this

method is always less efficient than using all the events with their corresponding weights.[3] A method of obtaining the least upper bound is to perform a search in the (3n-4) dimensional cube for the maximum of the function $r(\phi)\, w(\phi) = f(x_1 \ldots x_i \ldots x_{3n-4})$, where n is the number of final-state particles and the x_i are the random numbers in the intervals (0,1). The function $f(x_1 \ldots x_i \ldots x_{3n-4})$ is obtained by generating an event corresponding to a specific set of x_i^s and evaluating $r(\phi)$ and $w(\phi)$ for the event.

SAGE provides a standard option for finding the least upper bound for the event weights, $r(\phi)\, w(\phi)$.[4] This maximum weight is obtained with the subroutine call

CALL MAXWT (M,NRAND,NPRINT,WTMAX)

and by coding a function subroutine

FUNCTION RATE(M).

1. Input Quantities

M=Index entered by user and simply passed on to RATE for possible use there.

NRAND = number of random numbers used in generating an event. SAGE uses 3N-4 random numbers, where N is the number of particles in the final state.

NPRINT = flag that controls printing of information about the step by step progress of the search. Every time the current iteration number is zero or an exact multiple of NPRINT or when convergence has been achieved, the value of RATE and the values of the random numbers are printed. If NPRINT is zero or negative this output is completely supressed.

2. Output Quantity

WTMAX = the maximum value found for $r(\phi)\, w(\phi)$.

In the user coded function subroutine, the user must generate an event with SAGE, evaluate his matrix element squared, $r(\phi)$, for that event and set

$$\text{RATE} = r(\phi)\, w(\phi)$$

where $w(\phi)$ is the first weight, w(1), returned by the call to WT (see section IV).

As noted above, SAGE uses 3N-4 random numbers to generate an event. If the user employs additional random numbers in the event generation, or if he generates his own decay angular distributions (see section III-E), or his own invariant masses, he must obtain the random numbers with the subroutine call

```
CALL RAND(R,NR)
```

where

NR = number of random numbers desired

R = array of random numbers returned by RAND.

If the user does use random numbers external to SAGE for generating the event, care must be taken in determining NRAND. If the external random numbers are in addition to those used in SAGE then their number is added to the 3N-4 used by SAGE. If the external random numbers are used instead of those used in SAGE then they are already included in the 3N-4, and should not be added to that number. Examples of the latter case are user generated decay angular distributions or user generated invariant mass distributions, with the ISW=2 option for generating the four-vectors (see section III-D).

VIII. Statistical Accuracy and Generation Efficiency

The Monte Carlo method consists of generating a sample of N random events in the phase space volume v according to a normalized density, $f(\phi)$, and averaging $r(\phi)$, for these events. Then

$$R = (1/N) \sum_{i=1}^{N} r(\phi_i)/f(\phi_i), \qquad (6)$$

where ϕ_i is the phase space point corresponding to the ith random event. The statistical uncertainty in this evaluation is $\delta R = \sigma(r/f)N^{-1/2}$, where $\sigma(r/f)$ is the root-mean-square deviation of $r(\phi)/f(\phi)$ from its average value, R.

For a given number of Monte Carlo events the statistical uncertainty, δR, of the integration depends critically upon the phase space density distribution of the events, $f(\phi)$. The more closely $f(\phi)$ resembles $r(\phi)$ the higher the accuracy for the same number of Monte Carlo events. The accuracy of a Monte Carlo integration is usually characterized by its efficiency, ϵ, which is defined as R^2 defined by the average of $[r(\phi)/f(\phi)]^2$. This efficiency is equal to one for the case $f(\phi) = r(\phi)$ and is smaller the more $f(\phi)$ deviates from $r(\phi)$. In terms of the efficiency, the fractional statistical uncertainty in the Monte Carlo integration is given by

$$(\delta R/R) = (1/\epsilon - 1)^{\frac{1}{2}} N^{-\frac{1}{2}} \qquad (7)$$

The computational time required to evaluate Monte Carlo integrals grows linearly with the number of events required. Therefore, to minimize this time, the efficiency should be as large as possible.

The statistical accuracy of a Monte Carlo calculation can itself be calculated by Monte Carlo methods if it is not too small. The actual events used in the Monte Carlo calculation may also be used to evaluate ϵ provided that $\epsilon \gg 1 / N$. If this condition is not met then the value obtained for

ξ from the generated events may have no meaning, and calculations that appear to have reasonable statistical accuracy will be incorrect.

SAGE has a standard option for choosing the optimum values of the parameters that characterize phase space density distributions $f(\phi)$, in order to maximize the generation efficiency. Let α represent a set of parameters that govern a phase space density distribution (for example the parameter R in GENIS or the four parameters PARM(1) through PARM(4) in GODEL). If the efficiency is much larger than 1/N it can be evaluated by

$$\xi(\alpha) = \left[\sum_{i=1}^{N} r(\phi_i)/f(\alpha,\phi_i)\right]^2 \Big/ N \sum_{i=1}^{N}\left[r(\phi_i)/f(\alpha,\phi_i)\right]^2 \quad (8)$$

where $r(\phi_i)$ is the value of the users matrix element squared for the ith generated event and $1/f(\alpha,\phi_i)$ is the weight assigned by the event generator for the ith event. The best values for the parameters α are those that maximize $\xi(\alpha)$.

SAGE finds the best values for the parameters, α, by performing a search[4] in the space of the parameters for the maximum of $\xi(\alpha)$, Eqn. 8. This maximazation is obtained with the subroutine call

CALL MAXEFF(M,NPARM,NPRINT,PARM,DPARM,EFFMAX)

and by coding a function subroutine

FUNCTION EFF(M,NPARM,PARM)

1. Input Quantities

M = Index entered by user and simply passed on to EFF for possible use there.

NPARM = number of parameters to be varied

NPRINT = same meaning as in MAXWT (see section VIII)

PARM(NPARM) = array of starting values for the parameters

DPARM(NPARM) = characteristic distance or interval for each of the parameters. Used in the maximazation to help determine step distance for each iteration. (See Ref. 6)

2. Output Quantities

PARM(NPARM) = array of the solution values for the varied parameters

EFFMAX = solution value for the maximum efficiency.[5]

In the user coded subroutine, the user must generate a set of N events with SAGE using the values stored in PARM for the phase space density distribution parameters. For each generated event, the user's matrix element squared $r(\phi)$ must be evaluated and the sums indicated by Eqn. 8 must be performed. The user then sets EFF equal to the calculated efficiency for that set of N events and the current values of the parameters in PARM.

The optimum number of Monte Carlo events, N, used for the evaluation of ϵ in the search, depends upon several factors. The larger N, the more accurately the solution to the search will represent the best efficiency, and the solution values of the parameters will be the best ones for the Monte Carlo integration. However, the computational time required for the search increases linearly with increasing N. Thus, the time required for the search must be balanced with the computational time required for the ultimate generation of events for the integration of $r(\phi)$. This latter time increases linearly with decreasing efficiency. Empirically, it has been found that the choosing of N, such that $N\epsilon \leq 100$ gives an adequate estimation of the best values for the parameters.[5]

IX. Running Times

The computational time required by SAGE for event generation depends upon the user's phase space density distribution and which of the various SAGE options the user employs. It is impossible to present a complete set of

algorithums for all possible options, however, the following formulae may be useful.

The computational time required by RY and GQ to generate an event corresponding to uniform phase space density on a C. D.C. 6600 computer is

$$t(\text{msec}) = .56\,\theta(n,5)\,(n-4) + .48\,\theta(n,4) + .47\,\theta(n,3) + .23 \quad (9)$$

where n is the number of final state particles and $\theta(n,m)$ has the value zero for n less than m and one for n greater than or equal to m. For other frequency distributions additional computer time is required. For each system that is generated with an e^{at} or $1/(t-a^2)^2$ frequency .27 msec must be added to Eq. 9. The time added to Eq. 9 for each Breit-Wigner invariant mass generation depends upon n. For n=4, .19 msec is added and for n= 5, .12 msec is added for each Breit-Wigner generation.

Running times for GENIS are given in Ref. 4.

Computation time can be saved by the procedure employed in generating the events. If selections are to be made on certain dynamical variables, a generation procedure should be employed generating these variables first. Then if the event fails the selection a new event can be started without completing the generation of the failing event. Note, however, that a call to WT (or INITL or INITLØ) must be made before starting the new event. Similarly if less than the total number of 4-vectors are needed, the vertices should be arranged so that these 4-vectors are generated first; then only the masses of the remaining vertices need to be generated.

X. Some Examples

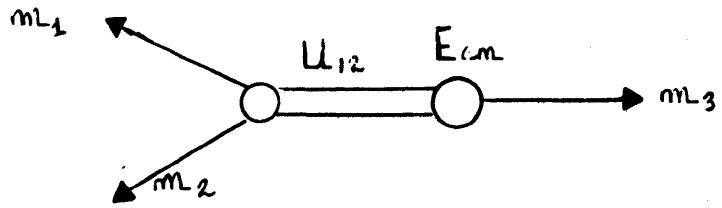
This section contains sample coding of SAGE event types for certain typical applications. These examples will illustrate a few of the techniques involved in using SAGE.

Consider the reaction, Beam + Target $\rightarrow A_1 + A_2 + A_3$. The following coding will generate a Monte Carlo event for this reaction with uniform phase space density.

```
      REAL S1(2), S2(2),S3(2),S4(2),P12(4,2),P1(4,2),P2(4,2),Pe(4,2),W(2)
      DATA X,S1,S2,S3,S4 /3HEND,m12,2.Ø,m3,1Ø,m1,1.Ø,m2,1.Ø
    1 CALL INITLØ(ECM)          (need execute only once)
      CALL RØ(S1,S2,ECM)
      CALL GO(P12,P3,U12,X)
      CALL RY(S3,S4,U12,P12,X)
      CALL GO(P1,P2,X)
      CALL WT(W)
```

In these examples lower case indicates numerical values inserted by the user.

The four vectors of A_1, A_2, and A_3 will be stored in P1, P2, and P3, respectively, and the phase space weight is stored in W. This procedure for generating the final state can be diagrammed as follows:



An alternate coding would be:

```
DIMENSION N(2),S(3),U1(2),P1(4,3),W(2)
DATA N,S /3,1,m1,m2,m3/

CALL INITLØ(ECM)        (need execute only once)

CALL GOGEN(ECM,2HCM,N,S,U1,P1)
CALL WT(W)
```

In the above examples the four vectors are generated in the center of mass frame.

If only the four vector of A_3 (or $A_1 + A_2$) is needed, the last three statements of the first example could be changed to

```
CALL RØ(S3,S4,U12)
CALL GØ(X)
CALL WT(W)
```

in order to save computational time. If only the invariant mass U12 is needed then the preceding statement could also be changed to

```
CALL GØ(U12,X)
```

or, in the second example, the sixth argument in the call to GOGEN could be changed to 3HEND.

If a cut is being made on the energy of the third particle, then the following statement could be placed between the fifth and sixth statements

of the first example:

```
IF(P3(4).GT.2.Ø) GO TO 1
```

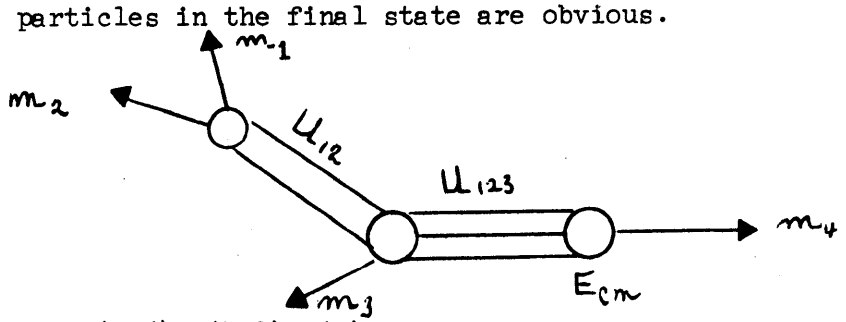
This suppresses the generation of extra four vectors for discarded events.

If the invariant mass U12 is to be given a Breit-Wigner frequency distribution and A_3 is to be given an e^{8t} distribution with respect to the target and A_1 an e^{4t} distribution with respect to the beam, the following coding could be used:

```
REAL S(2,4),E(4,2),D(3,2),P(4,6),W(2)
DATA S /m12,2.Ø,m3,1.Ø,m1,1.Ø,m2,1.Ø/
DATA E,D /.892,.Ø5,6*Ø.Ø,4HTDEL,2.Ø,8.Ø,4HBDEL,1.Ø,4.Ø/

CALL INITL(.93826,.4937,ECM,5.Ø,3HLAB)

CALL RY(S,S(1,2),ECM,3HLAB,E,D)
CALL GO(P,P(1,5),U12,3HEND)

CALL RY(S(1,3),S(1,4),U12,P,E(1,2),D(1,2))
CALL GO(P,P(1,3),3HEND)

CALL WT(W)
```

For this example the four vectors of A_1, A_2, and A_3, expressed in the laboratory frame, are stored in P(4,1),P(4,3), and P(4,5) respectively.

The following sample coding shows two ways of generating a four particle final state, Beam + Target $\rightarrow A_1 + A_2 + A_3 + A_4$ for uniform phase space generation. The extentions to Breit-Wigner mass and e^{at} (or cos θ) distributions as well as more particles in the final state are obvious.

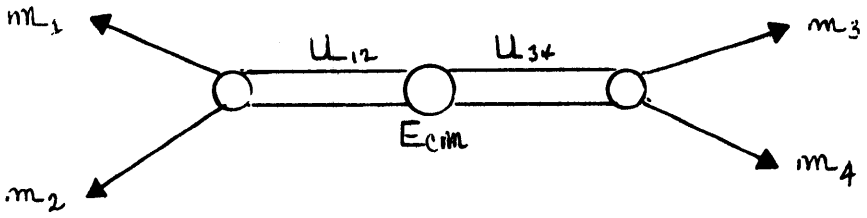



```
REAL S(2,6),P(4,8),W(2)
DATA S /m123,3.Ø,m12,2.Ø,m1,1.Ø,m2,1.0,m3,1.Ø,m4,1.Ø/

CALL INITLØ(ECM)

CALL RØ(S,S(1,6)ECM)
CALL GO(P,P(1,7).U123,3HEND)
```

```
      CALL RY(S(1,2),S(1,5),U123,P,3HEND)
      CALL GO(P,P(1,5),U12,3HEND)

      CALL RY(S(1,3),S(1,4),U12,P,3HEND)
      CALL GO(P,P(1,3),3HEND)

      CALL WT(W)
```

This example could also be coded as follows:

```
      DIMENSION S(4),N(2),W(2),U(3),P(4,4)
      DATA S,N /m1,m2,m3,m4,4,1/

      CALL INITLØ(ECM)

      CALL GOGEN(ECM,2HCM,N,S,U,P)

      CALL WT(W)
```




```
      REAL S(2,6),P(4,8),W(2)
      DATA S /m12,2.Ø,m34,2.Ø,m1,1.Ø,m2,1.Ø,m3,1.Ø,m4,1.Ø/

      CALL INITLO(ECM)

      CALL RØ(S,S(1,2),ECM)
      CALL GO(P,P(1,5),U12,U34)

      CALL RY(S(1,3)S(1,4),U12,P,3HEND)
      CALL GO(P,P(1,3),3HEND)

      CALL RY(S(1,5),S(1,6)U34,P(1,5),3HEND)
      CALL GO(P(1,5),P(1,7),3HEND)

      CALL WT(W)
```

Footnotes

1. The Breit-Wigner function used in SAGE is $BW(\mu) = 1/\{[(\mu-E)2/\Gamma]^2+1\}$ Here μ is the variable invariant mass and E and Γ are the parameters of the function, giving the central value and the full width at half maximum respectively.
2. The P = 3HIAB option is not available in GENIS.
3. Here the efficiency is defined by Eqn. 8. However, if the Monte Carlo events are processed further after they are generated and the matrix element squared, $r(\phi)$, evaluated, then it may in fact be more economical to use unweighted events.
4. SAGE employs the computer code MINF68 [6] for minimizing (or maximizing) a function of several variables.
5. For each evaluation of the efficiency SAGE reinitializes the random number generator to the same starting point. However, for the evaluation of EFFMAX SAGE uses a different set of random numbers for the evaluation of the efficiency. Thus, one can compare the value of EFFMAX returned by MAXEFF with the value of the efficiency printed by MAXEFF with the NPRINT option, at the solution. If the values are similar then the solution to the search probably represents the true efficiency for the parameters. If not, then more events should be used in the evaluation of the efficiency for the search.

References

1. L. Champomier, SUMX - A Data Summarizing Program, Lawrence Radiation Laboratory Report No. UCRL 11222, April 1964 (unpublished).

2. J.H. Friedman and A.R. Rittenberg, KIOWA - A General Description, Lawrence Radiation Laboratory Group A Programming Note No. P-171, May 1968 (unpublished).

3. J.H. Friedman, Journal of Comp. Physics, 7, 1 (1971).

4. L. Van Hove, Nucl, Phys., B9, 331 (1969); W. Kittel, L. Van Hove and W. Wojcki, CERN Report No. CERN/DPhII/Phys 70-8, 1970.

5. J.H. Friedman, G.R. Lynch, C.G. Risk, and T.A. Zang, Jr., An Efficient Monte Carlo Event Generation Method for Multiperipheral Models, Lawrence Radiation Laboratory Report No. UCRL 20141, November 1970; published in Journal of Comp. Physics, 8, 1 (Aug. 1971).

6. S. Derenzo, MINF68 - A General Minimizing Routine, Lawrence Radiation Laboratory, Group A Programming Note No. P-190, 1969 (unpublished).