JOURNAL OF COMPUTATIONAL PHYSICS 8, 144–152 (1971)

An Efficient Monte Carlo Event Generation Method for Multiperipheral Models

JEROME H. FRIEDMAN, GERALD R. LYNCH, CLIFFORD G. RISK, AND THOMAS A. ZANG, JR.*

Lawrence Radiation Laboratory, University of California, Berkeley, California 94720

Received November 17, 1970

A Monte Carlo event generation method is presented that integrates multiperipheral models with good efficiency for all energies and all multiplicities up to 18.

The multiperipheral model [1] has been the subject of considerable theoretical investigation in recent years. In order to predict experimental results from this model it is necessary to perform phase space integrals of the form $R = \int_v r(\phi)\, d^n\phi$, where ϕ is a point and $d^n\phi$ a volume element in the $(3n - 4)$-dimensional phase space, and v is the total phase space volume accessible to the final state. The integrand $r(\phi)$ is the transition matrix element squared calculated from the model. Because of the complexity of these phase space integrals, Monte Carlo techniques are often employed for their evaluation.

Detailed descriptions of Monte Carlo event generation are discussed elsewhere [2, 3] and only some of the basic concepts are discussed here. The Monte Carlo method consists of generating a sampe of N random events in the phase space volume v according to a normalized frequency $f(\phi)$ and, averaging $r(\phi)/f(\phi)$ for these events. Then

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

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.

There are several variations of the multiperipheral model, but they all have in common the property that the four-momentum transfers squared from the beam

* Present address: Physics Department, Notre Dame University, Notre Dame, Ind. 46556.

(or target) particle to certain final-state particle combinations are severely limited. This property is summarized by the transition matrix element squared:

$$r(\phi) = \exp\left(\sum_{i=1}^{n-1} a_i t_i\right), \tag{2}$$

where the t_i are the $(n-1)$ four-momentum transfers squared that are limited by the model and the a_i parametrize the degree of limitation. The particular t_i that the multiperipheral model limits are most easily described by the use of the multiperipheral diagram of Fig. 1. The two lines on the left represent the momenta

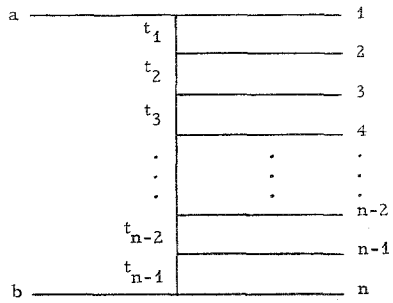

FIG. 1. A multiperipheral graph for an n-particle final state (see text).

of the two incident particles, and the lines on the right represent the momenta of the n final-state particles. The vertical lines linking the final-state particles represent the four-momentum transfer squared from incident particle a to the combination of particles above each link. These are the four-momentum transfers squared that appear in Eq. (2). Note that by momentum conservation these t_i are also the four-momentum transfers squared from incident particle b to the particle combination below each link.

For a given number of Monte Carlo events the statistical uncertainty δR of the integration depends critically upon the phase space frequency 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 divided 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

$$(\delta R/R) = (1/\epsilon - 1)^{1/2} N^{-1/2}. \tag{3}$$

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.

Early Monte Carlo event generators [2, 4] generated events with nearly constant phase space frequencies. These generators have good efficiency for integrating a constant matrix element squared, $r(\phi) = 1$, but they have a very low efficiency for integrating the multiperipheral model. This is because the model severely limits the t_i in Eq. (2), while the generator produces events whose t_i are roughly evenly distributed in the entire kinematic range.

Recently there have been several methods developed that improve the efficiency of Monte Carlo integrations for models of high-energy collisions [3, 5–7]. Noting that an experimental property of high-energy collisions is the limiting of the final-state particles' momenta transverse to the beam direction, Pene and Krzywicki [6] as well as Kittel, Van Hove, and Wojcik [7] have developed an event generator with a phase space frequency distribution that can be approximated by

$$f(\phi) = \exp\left(-b \sum_{i=1}^{n} p_i^2\right). \tag{4}$$

The p_i are the components of the final-state particles' momenta transverse to the beam direction. This generator integrates several models of high-energy collisions with good efficiency. However, most multiperipheral models concentrate the longitudinal momentum of the final-state baryons at large values, whereas this generator produces them with a more or less constant phase space distribution. This results in low efficiency for integrating these multipheripheral models.

Adressing themselves specifically to multipheripheral models, Byckling and Kajantie [5] have developed a Monte Carlo event generator that produces events with a phase space frequency that can be approximated at high energy by

$$f(\phi) = \prod_{i=1}^{n-1} \mu_{i+1} \exp a_i[(t_i - t_i^+)]. \tag{5}$$

Here the t_i are the same as those appearing in Eq. (2). The t_i^+ are the maximum values for each corresponding t_i. These maximum values vary from event to event and depend upon the values assigned to the integration variables generated before the specific t_i. Each μ_i is the invariant mass of the final-state particle combination for which t_i is the four-momentum transfer squared. This generator integrates the multiperipheral model of Eq. (2) with good efficiency as long as the energy is not too high or the multiplicity (numer of final-state particles, n) is not too large. Table Ia shows the efficiency of this generator for various multiplicities and laboratory beam momenta, for the reaction $pp \to pp(n-2)\pi$, where the a_i in both Eq. (2) and Eq. (5) are all taken to be 4.0.

This report presents a Monte Carlo event generation method that integrates

TABLE I

Efficiency of Monte Carlo integration for the matrix element squared $r(\phi) = \exp(4\sum_{i=1}^{n-1} t_i)$ as a function of the laboratory beam momentum, P, and final state multiplicity, n, in the reaction $pp \to pp(n-2)\pi$:
(a) for the generation method of Refs. [3, 5];
(b) for the generation method of this report.

n		$P \to$ 50 GeV/c	200 GeV/c	1000 GeV/c
4	(a)	0.20	0.14	0.08
	(b)	0.83	0.86	0.80
8	(a)	0.002	0.0007	0.0002
	(b)	0.41	0.43	0.43
16	(a)	~0.00005	<0.00002	<0.00002
	(b)	0.06	0.07	0.09

multiperipheral models with good efficiency for all energies and all multiplicities up to 18. Table Ib shows the efficiencies achieved with this generator for the same conditions discussed above for Table Ia. Other multipheripheral models can be integrated with similar efficiencies.

In order to understand this generation method it is necessary to understand the reasons why the frequency distribution of Eq. (5) loses efficiency with increasing energy and multiplicity. This is most easily done by first considering the special case in which all of the particles in the reaction have zero rest mass. For this case one has

$$t_i^+ = t_{i+1}(\mu_i/\mu_{i+1})^2, \tag{6}$$

with $t_n = 0$. Inserting this into Eq. (5) ,one has

$$f(\phi) = \prod_{i=1}^{n-1} \mu_{i+1} \exp\{a_i[t_i - t_{i+1}(\mu_i/\mu_{i+1})^2]\} \tag{7a}$$

or (neglecting constant factors)

$$f(\phi) = \exp(a_1 t_1) \prod_{i=2}^{n-1} \mu_i \exp\{[a_i - a_{i-1}(\mu_{i-1}/\mu_i)^2]\, t_i\}. \tag{7b}$$

The $n-2$ invariant masses are generated first with frequency

$$f(\mu) = \prod_{i=2}^{n-} , \tag{8}$$

and then, using these invariant masses, the $n-1$ four-momentum transfers squared are generated with the frequency

$$f(t) = \exp(a_1 t_1) \prod_{i=2}^{n-1} \exp\{[a_i - a_{i-1}(\mu_{i-1}/\mu_i)^2]\, t_i\}. \tag{9}$$

Generation of the invariant masses with the frequency of Eq. (8) yields

$$\langle(\mu_{i-1}/\mu_i)^2\rangle = 1 - (2/i). \tag{10}$$

Replacing $(\mu_{i-1}/\mu_i)^2$ by this average value in the exponent of Eq. (9) and letting all of the a_i have the same value a, one has

$$f(\phi) = \exp(at_1) \prod_{i=2}^{n-1} \mu_i \exp[(2/i)\, at]. \tag{11}$$

By comparing this frequency distribution with the integrand, $r(\phi)$, of Eq. (2) the reasons for its loss of efficiency with increasing multiplicity becomes apparent. The less dramatic loss of efficiency with increasing energy is not illustrated because of the approximations employed in obtaining Eq. (11). Compared with the multiperipheral model of Eq. (2), the frequency $f(\phi)$ of Eqs. (7) and (11) overpopulates large values of invariant masses μ_i, while underpopulating the regions of low-four-momentum transfer squared t_i. From Eq. (11) it is clear that this trend is enhanced with increasing multiplicity. The Monte Carlo event generator presented here overcomes these difficulties by concentrating the invariant masses μ_i at low values and then generating each of the four-momentum transfers squared t_i so that they more closely resemble the distributions predicted by the multiperipheral model.

The invariant masses are generated first, in order, starting with the two-particle system μ_2. That is,

$$\mu_2 = (E - S_n)\, \rho_2 + S_2, \tag{12}$$

where E is the center-of-mass energy of the reaction, S_n the sum of the final-state particles' rest masses, and S_2 the sum of the rest masses of the particles that compose μ_2. The dimensionless quantity ρ_2 is a random variable generated in the interval 0 to 1. The other invariant masses are obtained by

$$\mu_i = (E - S_n + S_i - m_i - \mu_{i-1})\, \rho_i + \mu_{i-1} + m_i. \tag{13}$$

Here m_i is the mass of the ith particle and $S_i = \sum_{j=1}^{i} m_j$. Equations (12) and (13) ensure energy conservation. The frequency distributions of the generated random variables $h_i(\rho_i)$ are arbitrary, but the choice determines the frequency distribution of the generated invariant masses and thus effects the efficiency of the integration

of a particular model $r(\phi)$. The methods employed by previous Monte Carlo generators [2–6] for generating the invariant masses μ_i are equivalent to

$$h_i(\rho_i) = (1 - \rho_i)^{n-i-1}. \tag{14}$$

As illustrated above for the zero-rest-mass case, this leads to an overpopulation of large values for these invariant masses when integrating the multiperipheral model.

In order to increase the population of low invariant mass events this generator employs the frequency

$$h_i(\rho_i) = \exp(-b_i\rho_i). \tag{15}$$

The $n - 2$ parameters b_i are chosen so that the resulting invariant mass distributions resemble as closely as possible those predicted by $r(\phi)$, the model to be integrated. For the multiperipheral model, Eq. (2), it can be shown [1] that

$$\langle \mu_i \rangle \sim E^{(i-1/n-1)}. \tag{16}$$

Neglecting the rest masses of the final-state particles and assuming most of the invariant masses have small values, one has approximately $\mu_i = E\rho_i$, so that

$$h_i(\rho_i) = \exp(-b_i\mu_i/E), \tag{17}$$

for which

$$\langle \mu_i \rangle = E/b_i. \tag{18}$$

Comparing Eqs. (16) and (18), one finds

$$b_i \sim E/\langle \mu_i \rangle \sim E^{1-(i-1/n-1)} \sim E^{(n-i/n-1)}. \tag{19}$$

Using this result as a hint, we parametrize

$$b_i = b_0(E/E_0)^{(n-i/n-1)} - 1. \tag{20}$$

The values of the two parameters, b_0 and E_0, are chosen so that the resulting $n - 2$ invariant masses are generated as closely as possible to those predicted by the multiperipheral model. This can be conveniently accomplished by generating a sample of N events and choosing those values of E_0 and b_0 that minimize the fractional statistical uncertainty in the Monte Carlo integration of $r(\phi)$. From Eq. (3) it is seen that this is equivalent to maximizing the efficiency ϵ for the integration.

This maximization can be carried out by performing a search in the two-dimensional space for the maximum of the function

$$\epsilon(b_0, E_0) = \Big[\sum_{i=1}^{N} r(\phi_i)/f(\phi_i)\Big]^2 \Big/ N \sum_{i=1}^{N} [r(\phi_i)/f(\phi_i)]^2. \tag{21}$$

Here $r(\phi_i)$ is the value of the matrix element squared predicted by the model for the ith event and $1/f(\phi_i)$ is the weight assigned by the event generator the ith event. The search for the optimum values of E_0 and b_0 can be performed by using one of the many computer codes that otpimize a function of several variables. Using MINF68 [8], acceptable values for the parameters are usually obtained after four or five iterations. Since only relative values of $\epsilon(b_0, E_0)$ are important in the search, the random number generator employed by the event generator should be initilized to the same stating point for each evaluation of ϵ.

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 $N\epsilon \gtrsim 100$ gives an adequate estimation of the best values for the parameters.

After generating the invariant masses, the four-momentum transfers squared t_i are generated with the frequency

$$f(t) = \exp\left[\sum_{i=1}^{n-1} d_i(t_i - t_i^+)\right]. \tag{22}$$

The $(n-1)d_i$ are chosen so as to populate the t_i as closely as possible to those predicted by the multiperipheral model Eq. (2). The Monte Carlo event generators described in Refs. [3, 5] used $d_i = a_i$ for all values of i. For the zero-rest-mass case Eqs. (9) and (10) indicate that

$$d_i = a_i/[1 - (a_{i-1}/a_i)(1 - 2/i)], \qquad d_1 = a_1 \tag{23}$$

would be a better choice. In general Eq. (9) suggests that

$$d_i = a_i/[1 - (a_{i-1}/a_i)\langle(\mu_{i-1}/\mu_i)^2\rangle], \qquad d_1 = a_1 \tag{24}$$

is a good prescription, where the average $\langle(\mu_{i-1}/\mu_i)^2\rangle$ is evaluated for the frequency distribution of the generated invariant masses. For the invariant mass distributions implied by Eqs. (12), (13), (15), and (20) this average value is difficult to evaluate. Therefore we simply take

$$d_i = d_1 + d'(i - 1). \tag{25}$$

The best values of d_1 and d' can be obtained by searching for the maximum of

the efficiency, Eq. (21), in the space of these parameters. Since the best values of these parameters depend upon the generation frequencies of the invariant masses, they will depend upon the values used for b_0 and E_0, Eq. (20). Therefore, the optimization for b_0 and E_0 should be performed before the optimization for d_1 and d'. Another procedure is to search for the maximum of the efficiency in the space of all four parameters simultaneously.

To the extent that the approximations leading to Eqs. (20) and (25) are satisfied, the four parameters b_0, E_0, d_1, and d' should be independent of energy and multiplicity. Table II gives the solution values of these parameters for the same conditions described for Table I. Inspection of Table II shows that each of the parameters has a slow dependence on energy and a bit more rapid dependence on multiplicity.

TABLE II

Values of the parameters b_0, E_0, d_1, and d' (see text) that yielded the efficiencies of Table Ib.

n		$P \to$ 50 GeV/c	200 GeV/c	1000 GeV/c
	b_0	2.4	2.2	1.8
4	E_0	1.1	1.3	1.8
	d_1	3.4	3.7	3.7
	d'	0.62	0.39	0.34
	b_0	3.2	2.9	2.4
8	E_0	0.58	0.67	0.82
	d_1	4.7	4.8	4.7
	d'	0.72	0.44	0.30
	b_0	6.0	6.0	3.7
16	E_0	0.34	0.48	0.41
	d_1	8.6	8.7	7.9
	d'	0.63	0.29	0.25

The procedures and parametrization discussed above are not limited to a specific multiperipheral model [for example, Eq. (2)]. Any model with similar characteristics can be integrated with comparable efficiency. The best values of the parameters b_0, E_0, d_1, and d' can be found for any $r(\phi)$ by inserting it into Eq. (21) and solving for the maximum of the efficiency in the space of the parameters. Also, one can alter the parametrization itself for sufficiently different models.

REFERENCES

1. N. F. BALI, G. F. CHEW, AND A. PIGNOTTI, *Phys. Rev.* **163** (1967), 1572.
2. F. JAMES, CERN Yellow Report No. 68-15, 1968 (unpublished).
3. J. FRIEDMAN, *J. Comput. Phys.* **7** (1971), 201.
4. G. LYNCH, OWL, A Monte Carlo phase space program, Lawrence Radiation Laboratory Group A Programming Report No. P-162, 1967 (unpublished).
5. E. BYCKLING AND K. KAJANTE, *Nucl. Phys.* **B9** (1969), 568; E. BYCKLING, M. KAARTINEN, K. KAJANTIE, AND H. VILLANEN, *J. Comput. Phys.* **4** (1969), 521; E. BYCKLING AND K. KAJANTE, *Phys. Rev.* **187** (1969), 2008.
6. O. PENE AND K. KRZYWICKI, *Nucl. Phys.* **B12** (1969), 415.
7. L. VAN HOVE, *Nucl. Phys.* **B9** (1969), 331; W. KITTEL, L. VAN HOVE, AND W. WOJCIK, CERN Report No. CERN/D PhII/Phys 70–8, 1970.
8. S. DERENZO, MINF 68-A general minimizing routine, Lawrence Radiation Laboratory Group A Programming Note No. P-190, 1969 (unpublished).