CERN-TH.6155
July 1991

Monte Carlo exponentiation and final-state radiative corrections

Ronald Kleiss
TH Division, CERN, Genève

Abstract

It is argued that final-state radiative corrections to many LEP processes, although themselves numerically small under many experimental conditions, are essential ingredients in any Monte Carlo treatment that aims at a sufficiently accurate description of angular distributions as well as total cross sections. A class of Monte Carlo algorithms is presented which allow for full-fledged exponentiation including the emission of arbitrarily large numbers of bremsstrahlung photons. Special attention is given to the convergence properties of the algorithms for large photon multiplicities.

CERN-TH.6155/91
July 1991

1 Introduction

With the commencement of high-precision phenomenology at LEP, high-order QED effects have become important, in particular those of the resummation of soft-photon corrections to all orders of perturbation theory: the so-called exponentiation [1, 2]. Numerically the most important of these are the QED corrections to the initial-state, at least at the Z peak where σ_h^0 , the peak cross section for hadron production, is reduced by 30% to first order in α : upon exponentiation this becomes 26% [3, 4]. These effects are much larger than the experimental accuracy, which is about 1% or better. Fortunately the theory of these QED corrections is very well established and one can in principle extend the precision of the predictions to a sufficient degree that the QED effects can be unfolded from the data to yield information on the theoretically more interesting weak effects. The calculations of the QED corrections to the line shape have been carried out by a number of groups [4, 5, 6, 7] with a variety of different methods: the results agree to within 0.2%.

If one goes beyond the line shape to more exclusive quantities such as angular distributions and angular asymmetries, the situation becomes gradually less satisfactory. The forward-backward asymmetry has also been computed by various groups [9]: the agreement on the QED effects is satisfactory at the Z peak but deteriorates quickly if one moves away from the peak. In particular the use of the same structure functions for the line shape and for the forward-backward asymmetry, as in [8], has been disputed in [10].

On the side of Monte Carlo approaches a number of ways have been proposed to tackle the QED effects [11]. Some authors [12, 13] have modified the single-bremsstrahlung energy spectrum in existing one-loop Monte Carlo programs such as [14, 15, 16, 17], so as to approximate the energy loss to bremsstrahlung as predicted by exponentiated QED; however in events generated by such programs at most only one photon can occur, so that while some total rates can be predicted correctly, the full event topology cannot be described: for instance, aplanar events will not be generated. Monte Carlo programs based on the structure function approach [18, 19, 7] do in fact allow for the emission of more than single photons, but their correctness can so far only be proved for the total cross section: in other words, aplanar events do occur in the generated event sample but it is not known whether the programs predict the correct aplanarity distribution. Also, as mentioned above it has been argued [10] that structure functions which give the correct total cross section do not necessarily also produce the correct forward-backward asymmetry. In fact it has been proven rigorously [20] that the results of any structure-function Monte Carlo have an intrinsic limitation on their accuracy, and that no existing structure-function based Monte Carlo comes even close to that theoretical limit.

It is only the work of Jadach and Ward[21] that has any claim to be close to an exact, full-fledged QED exponentiation à la Yennie, Frautschi and Suura (YFS)

[22] including arbitrary numbers of (in principle) correctly distributed bremsstrahlung photons. Their program KORALZ [23], however, contains only exponentiation of initial-state radiation: final-state QED corrections are included up to one-loop accuracy only, so that only one photon at most is emitted by the produced fermions. An analogous program BHLUMI [24] attempts to implement the YFS scheme in Bhabha scattering.

The main reason why final-state radiation effects, and the interference between initial- and final-state radiation, have so far not been included in any exponentiation scheme is that they often happen to be numerically small.

The interference is a small effect because when the Z^0 is produced close to its mass shell, the wave functions for radiation from the initial and from the final state become separated in space-time due to the small but finite Z^0 lifetime, and their overlap decreases: for loose experimental cuts the interference is typically about 10^{-3} or smaller [25]. Whenever tight cuts are imposed or one moves away from the Z^0 peak, this argument is invalidated and the interference increases to its ‘natural’ value of several per cent.

Owing to the KLN theorem [26] the total final-state radiative correction is always small (0.17% for massless fermions with unit charge) but it can again become more sizeable for tight cuts: if a cut of s_0 is imposed on the mass squared M^2 of the produced fermion pair, the $\mathcal{O}(\alpha)$ final-state radiatively corrected cross section becomes

$$\begin{aligned}
\sigma(M^2 \geq s_0) &= \sigma_0(s) \left[1 + \frac{\alpha}{\pi} \left\{ \left(\log \left(\frac{s_0}{m_f^2} \right) - 1 \right) \left(\frac{s_0}{s} + \frac{s_0^2}{s^2} + 2 \log \left(1 - \frac{s_0}{s} \right) \right) \right. \right. \\
&\quad \left. \left. + \frac{3}{4} - \frac{s_0}{s} - \frac{s_0^2}{4s^2} + 2 \text{Li}_2 \left(\frac{s_0}{s} \right) \right\} \right] \\
&\sim \sigma_0(s) \left[1 + \frac{\alpha}{\pi} \left\{ \frac{3}{4} - \frac{s_0}{s} \left(\log \left(\frac{s_0}{m_f^2} \right) - 2 \right) \right\} \right] , \quad s_0 \ll s \\
&\sim \sigma_0(s) \left[1 + \frac{\alpha}{\pi} \left\{ 2 \log \left(1 - \frac{s_0}{s} \right) \left(\log \left(\frac{s}{m_f^2} \right) - 1 \right) \right. \right. \\
&\quad \left. \left. + \frac{3}{2} \log \left(\frac{s}{m_f^2} \right) + \frac{\pi^2}{3} - 1 \right\} \right] , \quad s_0 \sim s , \quad (1)
\end{aligned}$$

where s is the invariant mass of the incoming e^+e^- system (possibly after the emission of initial-state bremsstrahlung) and m_f is the mass of the produced fermions: $\text{Li}_2(x)$ is the dilogarithm function, $\text{Li}_2(x) = \sum_{k=1}^{\infty} x^k/k^2$. For a typical experimental cut $M > \sqrt{s}/2$ we find for muon pairs at the Z peak $\sigma(M^2 > s/4) \sim 0.995\sigma_0(s)$, i.e. a negative correction of about 0.5% which is easily comparable to the experimental accuracy.

2 Final-state exponentiation

From the above considerations it appears that there are at least three situations in which higher-order QED effects in initial-state radiation have to be supplemented by final-state radiation and the interference. Usually the final-state radiation effects themselves contribute significantly only at the $\mathcal{O}(\alpha)$ level at all energies. For tight experimental cuts with an effective s_0 close to s , or away from the Z^0 peak, higher-order interference effects are expected to become numerically significant. Since these interference effects increase in importance with decreasing fermion scattering angle θ they are especially relevant in small-angle Bhabha scattering – after all, it is precisely the interference terms that change the scale in the dominant leading logarithms in the QED corrections from s to t , where t is the momentum transfer in the spacelike channel:

$$\log\left(\frac{s}{m_e^2}\right) \rightarrow \log\left(\frac{s}{m_e^2}\right) + \log\left(\frac{1 - \cos\theta}{2}\right) = \log\left(\frac{-t}{m_e^2}\right). \quad (2)$$

Indeed, this is just a reflection of the fact that the theoretical notions of initial- and final-state radiation are not gauge-invariant in the t -channel scattering, and it is rather the combination of radiation from the incoming and outgoing electron (or positron) that can be meaningfully discussed – together, of course with their interference¹. This shows that in such situations the interference is a fundamental part of the QED corrections rather than an additional minor effect.

It appears that if one wants to simulate accurately the initial-final-state interference with exponentiation, then also final-state exponentiation has to be included in the simulation. This necessity, as is clear from the above, is not based on the numerical importance of the final-state radiative corrections themselves but rather on the conceptual necessity to have multiple photon emission from the final state in order to have a sensible discussion of the quantum interference effects between those Feynman diagrams where a given photon is emitted from the incoming fermion lines and those with a photon attached to the outgoing fermion lines. It is therefore imperative that in each generated Monte Carlo event with photons, each photon must have a finite probability of having been emitted (in the algorithm) by the initial-state fermions as well as a probability of having been emitted by the final-state fermions: if this is not the case (depending on the algorithm that was used), then the interference cannot be meaningfully defined, let alone simulated. This conceptual problem is at the basis of the fact that in the Monte Carlo programs KORALZ or DYMU2 the interference cannot be included, since every radiated photon is explicitly emitted either by the incoming or by the outgoing fermions. Indeed, in KORALZ it is possible to have either exponentiation of the initial-state radiation and no interference, or interference at the cost of no exponentiation (in this option the program simulates the $\mathcal{O}(\alpha)$ corrections only).

In the rest of this paper we shall study algorithms for exponentiated final-state

¹In the language of Jadach *et al.* this is called the ‘up-down’ interference.

radiation: as already mentioned, not because these effects are themselves large, but because it is a necessary ingredient of all fully exponentiated Monte Carlo programs. An implementation of the algorithms discussed here into an actual Bhabha scattering Monte Carlo will be discussed elsewhere [27].

Let us now briefly review the YFS exponentiation scheme for final-state bremsstrahlung photons. In order to regulate the infrared divergence inherent in any bremsstrahlung cross section we assume a cutoff energy such that bremsstrahlung photons with an energy lower than the cutoff are not detectable. Of course once we have included both soft and hard bremsstrahlung the final result should no longer depend on the cutoff value, both in analytical and Monte Carlo treatments. If we denote the cutoff by ϵ , then the bremsstrahlung spectrum can be described ² by

$$\begin{aligned} \frac{d\sigma}{\sigma_0} &\equiv \Psi(k) dk = \frac{4\pi^2}{k^0} V(\epsilon) \delta(k^0) + H(k) \theta(k^0 > \epsilon\sqrt{s}/2) dk , \\ V(\epsilon) &= \frac{\alpha}{\pi} \left[2 \log\left(\frac{1}{\epsilon}\right) \left(\log\left(\frac{s}{m_e^2}\right) - 1 \right) + \frac{3}{2} \log\left(\frac{s}{m_e^2}\right) - 1 + \frac{\pi^2}{3} \right] , \end{aligned} \quad (3)$$

where ϵ is implied to be very small. The photon four-momentum is k and $H(k)$ stands for the matrix element for single hard bremsstrahlung. The phase space integration element dk is an abbreviation for the actual one which reads $k^0 dk^0 d\Omega/(16\pi^3)$, where Ω is the photon solid angle. Note that for $\epsilon \rightarrow 0$ the contribution from virtual and soft photons, $V(\epsilon)$, goes to $-\infty$ so that in a Monte Carlo program (as opposed to an analytic calculation) ϵ cannot be taken to be arbitrarily small: both pieces of $\Psi(k)$ must be positive in order to allow for the probabilistic interpretation inherent in any Monte Carlo approach. In fact, if we restrict ourselves to $\mathcal{O}(\alpha)$ only, ϵ cannot be taken much less than about 10^{-2} . The total $\mathcal{O}(\alpha)$ correction is given by

$$\int \Psi(k) dk = V(\epsilon) + \int_{k^0 > \epsilon\sqrt{s}/2} H(k) dk = \frac{3\alpha}{4\pi} , \quad (4)$$

where the integral runs over the total admissible phase space.

We now turn to the multi-photon case. We shall assume that the higher-order corrections simply factorize into separate pieces for each photon. This assumption is of course not completely justified: there are non-factorizing contributions to the higher-order corrections. However, these are exactly the kind of effects that Monte Carlo can take into account in a fairly trivial manner, by assigning to the generated events an appropriate weight factor. We shall therefore not bother to calculate these effects explicitly in this paper. Neither shall we take into account the phase space limitations which make the available phase space for each photon in multiple bremsstrahlung smaller than that for single bremsstrahlung. Under these assumptions the

²Here and in the following, a θ function of an inequality is defined to equal 1 when the inequality is satisfied, otherwise 0.

total cross section σ consists of contributions from zero, one, two, three... photons, and we can write

$$\frac{\sigma}{\sigma_0} = 1 + \int \Psi(k) dk + \frac{1}{2!} \iint \Psi(k_1) \Psi(k_2) dk_1 dk_2 + \frac{1}{3!} \iiint \Psi(k_1) \Psi(k_2) \Psi(k_3) dk_1 dk_2 dk_3 + \dots, \quad (5)$$

where in each integration the phase space limitations are implied. The combinatorial factors $1/n!$ arise from the Bose symmetry of the photons. By collecting all terms containing one or more $V(\epsilon)$'s we derive immediately that

$$\frac{\sigma}{\sigma_0} = \exp(V(\epsilon)) \left[1 + \int H(k) dk + \frac{1}{2!} \iint H(k_1) H(k_2) dk_1 dk_2 + \frac{1}{3!} \iiint H(k_1) H(k_2) H(k_3) dk_1 dk_2 dk_3 + \dots \right]. \quad (6)$$

Note that owing to this rewriting all contributions to the cross section have become positive for any finite value of ϵ . In the usual soft-photon approximation one then proceeds to evaluate the multiple H integrals analytically, but Eq.(6) also allows a straightforward Monte Carlo implementation of the YFS scheme, as was first pointed out by Jadach and Ward [21]. In such an implementation one first decides on the photon multiplicity of the event that is to be generated, using the multiple integrals over the F 's as relative probabilities. One then generates the fermion and photon momenta. Finally one multiplies the total cross section by $\exp(V(\epsilon))$. In the approximation where the multiphoton integrals factorize, the photon multiplicity is distributed according to the Poisson distribution with mean value $\int F(k) dk$. Hence for decreasing ϵ the dominant contribution to the cross section comes from events with higher photon multiplicities, in accordance with the interpretation of ϵ as a kind of 'energy resolution' of the Monte Carlo simulation: with better resolution, lower photon energies are resolved, and more (soft) photons are seen to occur. However, an experimentally observable quantity like the multiplicity of photons *with a given minimum energy* must go to a limiting value for infinitesimally small ϵ .

3 Generating multiphoton final states

3.1 The generating algorithm

We now turn our attention to possible algorithms for generating final-state photon momenta in addition to fermions. We shall do this in the form of a *unitary algorithm*, that is a number of prescriptions (in analytic form) which transforms a set of random numbers into a set of final-state variables with unit probability. Let us suppose that in addition to the two fermion momenta a number n of final-state radiation photons is

generated, using N independent pseudo-random variables r_i ($i = 1, \dots, N$) uniformly distributed between 0 and 1. Working in the centre-of-mass frame with total invariant mass w a general unitary algorithm is then formally described by

$$\begin{aligned}
1 &= \int_0^1 \cdots \int_0^1 dr_1 dr_2 \cdots dr_N \\
&= \int \delta^3 \left(\vec{q}_1 + \vec{q}_2 + \sum_{j=1}^n \vec{k}_j \right) \delta \left(q_1^0 + q_2^0 + \sum_{j=1}^n k_j^0 - w \right) F(q_1, q_2, k_1, \dots, k_n) \\
&\quad \times \left[\prod_{i=1}^2 \frac{1}{(2\pi)^3} d^4 q_i \delta(q_i^2 - m^2) \right] \left[\prod_{j=1}^n \frac{1}{(2\pi)^3} d^4 k_j \delta(k_j^2) \right] , \quad (7)
\end{aligned}$$

where $q_{1,2}^\mu$ are the fermion momenta with mass m , and the k_j^μ are the massless photon momenta. The delta functions stand of course for conservation of energy and momentum. The function $F(q_1, q_2, k_1, \dots, k_n)$ is related to the Jacobian of the transformation (which, however, is not invertible if $N > 3n + 2$). In the following we shall call F the *local density* of the algorithm: it is the probability density, for a given set of momenta $q_1^\mu, q_2^\mu, k_1^\mu, \dots, k_n^\mu$, that the algorithm will yield this particular set. In usual practice it is relatively easy to construct an algorithm for making momenta that satisfy the energy-momentum conservation conditions, but harder to obtain an explicit formula for the local density, as we shall see.

Once the local density is given, any cross section can in principle be obtained by Monte Carlo: one simply assigns to each generated event a weight

$$W_{\text{MC}} = \frac{A(q_1, q_2, k_1, \dots, k_n)}{F(q_1, q_2, k_1, \dots, k_n)} , \quad (8)$$

where A is the exact matrix element for this final state, multiplied by $(2\pi)^4/(2s)$. Insertion of this weight into Eq.(7) immediately shows that in this way indeed the exact cross section is evaluated. Note that (as we shall indeed do in the following) A may be put to zero whenever the momenta attain some unwanted values without destroying the validity of this approach.

We now outline the algorithm. The we shall cast it in a suitable analytic unitary form and subsequently work out the local density formula.

1. We start by generating two fermion momenta $\hat{p}_1^\mu, \hat{p}_2^\mu$, satisfying the momentum conservation constraints by themselves. For simplicity we shall not assign any nontrivial angular distribution to these momenta (although of course, in particular for Bhabha scattering, this will be necessary: however it is irrelevant to the rest of this discussion). From the equality

$$\int \left[\prod_{i=1}^2 \frac{1}{(2\pi)^3} d^4 \hat{p}_i \delta(\hat{p}_i^2 - m^2) \right] \delta^3(\vec{\hat{p}}_1 + \vec{\hat{p}}_2) \delta(\hat{p}_1^0 + \hat{p}_2^0 - w) = \frac{\beta}{4(2\pi)^5} \equiv A_f , \quad (9)$$

where $\beta = \sqrt{1 - 4m^2/w^2}$, we now see that this part of the unitary algorithm corresponds to

$$1 = \frac{1}{A_f} \int \left[\prod_{i=1}^2 \frac{1}{(2\pi)^3} d^4 \hat{p}_i \delta(\hat{p}_i^2 - m^2) \right] \delta^3(\vec{\hat{p}}_1 + \vec{\hat{p}}_2) \delta(\hat{p}_1^0 + \hat{p}_2^0 - w) . \quad (10)$$

2. Secondly we generate n photon momenta. These are not to be generated isotropically because of the huge collinear peaks in the exact cross section. A simple and adequate approximation to the radiative cross section for photon j is proportional to

$$S(\hat{p}_1, \hat{p}_2, \hat{k}_j) = e^2 \frac{(\hat{p}_1 + \hat{p}_2)^2}{(\hat{p}_1 \cdot \hat{k}_j)(\hat{p}_2 \cdot \hat{k}_j)} , \quad (11)$$

where $\hat{k}_j^\mu$ is the photon momentum and e is the fermion charge. This is of course not the real dependence on $\hat{k}_j^\mu$: in fact using Eq.(11) we are even missing some infrared-divergent terms. This happens not to be a problem as is discussed in greater detail in appendix A. In addition to Eq.(11) we also introduce a regulator function $\psi(\hat{k}_j^0)$ which goes to zero sufficiently fast so that the integral over the photon phase space exists³:

$$\begin{aligned} A_\gamma &= \int \frac{1}{(2\pi)^3} d^4 \hat{k}_j \delta(\hat{k}_j^2) S(\hat{p}_1, \hat{p}_2, \hat{k}_j) \psi(\hat{k}_j^0) \\ &= \frac{2\alpha}{\pi} \log \left(\frac{1+\beta}{1-\beta} \right) \int_0^\infty dx \frac{1}{x} \psi(x) . \end{aligned} \quad (12)$$

Leaving aside the relatively trivial and rather technical problem of how to generate values of $\hat{k}_j^\mu$ according to the distribution (12), the part of the unitary algorithm corresponding to the photon momenta generation can be written as

$$1 = \frac{1}{A_\gamma^n} \int \left[\prod_{j=1}^n \frac{1}{(2\pi)^3} d^4 \hat{k}_j \delta(\hat{k}_j^2) S(\hat{p}_1, \hat{p}_2, \hat{k}_j) \psi(\hat{k}_j^0) \right] . \quad (13)$$

Explicit algorithms to obtain the photon energies from pseudorandom numbers are given in appendix B.

3. We can now define the total generated momentum

$$Q^\mu = \hat{p}_1^\mu + \hat{p}_2^\mu + \sum_{j=1}^n \hat{k}_j^\mu . \quad (14)$$

³We might have the regulator ψ depend also on the direction of the $\hat{k}_j^\mu$ as well as on its energy: this however would make the algorithm extremely difficult to analyze and we shall not adopt this option.

Of course we now no longer have $\vec{Q} = 0$ and $Q^0 = w$. Therefore we perform a boost and a scaling transform on the momenta such as to restore overall conservation of four-momentum. To this end we introduce

$$\begin{aligned}\vec{b} &= -\vec{Q}/\sqrt{Q^2} , \\ \gamma &= \sqrt{1 + \vec{b}^2} , \\ x &= w/\sqrt{Q^2} ,\end{aligned}\tag{15}$$

and perform on each momentum $p^\mu (= \hat{p}_i^\mu, \hat{k}_j^\mu)$ the following Lorentz boost and scaling transform:

$$\begin{aligned}p^0 &\rightarrow x H_{\vec{b}}(p)^0 \equiv x (\gamma p^0 + \vec{b} \cdot \vec{p}) , \\ \vec{p} &\rightarrow x \vec{H}_{\vec{b}}(p) \equiv x (\vec{p} + p^0 \vec{b} + \vec{b}(\vec{b} \cdot \vec{p})/(1 + \gamma)) .\end{aligned}\tag{16}$$

It is easily checked that under this transformation indeed $H_{\vec{b}}^0(Q) = w$ and $\vec{H}_{\vec{b}}(Q) = 0$. In fact one might choose instead of this conformal transformation another one, for instance a Galilei transform followed by a scaling, but the inherent Lorentz-covariance in the choice of Eq.(16) makes the derivation of the local density much simpler.

4. After the scaling transform, the fermion momenta of course have no longer mass m but rather mass xm . To restore the correct masses a *mass fixup transformation* is required. At high energies this is however only a minor effect and we defer its discussion to later on.

The complete unitary algorithm up to step (4) can now be summarized analytically as follows:

$$\begin{aligned}1 &= \frac{1}{A_f A_\gamma^n} \int \left[\prod_{i=1}^2 \frac{1}{(2\pi)^3} d^4 \hat{p}_i \delta(\hat{p}_i^2 - m^2) \right] \delta(\vec{\hat{p}}_1 + \vec{\hat{p}}_2) \delta(\hat{p}_1^0 + \hat{p}_2^0 - w) \\ &\quad \left[\prod_{j=1}^n \frac{1}{(2\pi)^3} d^4 \hat{k}_j \delta(\hat{k}_j^2) S(\hat{p}_1, \hat{p}_2, \hat{k}_j) \psi(\hat{k}_j^0) \right] \\ &\quad d^3 \vec{b} \delta \left(\vec{b} + \frac{\vec{Q}}{\sqrt{Q^2}} \right) dx \delta \left(x - \frac{w}{\sqrt{Q^2}} \right) \\ &\quad \left[\prod_{i=1}^2 d^4 p_i \delta^4(p_i - x H_{\vec{b}}(\hat{p}_i)) \right] \\ &\quad \left[\prod_{j=1}^n d^4 k_j \delta^4(k_j - x H_{\vec{b}}(\hat{k}_j)) \right] ,\end{aligned}\tag{17}$$

where the integral is understood to include all integration elements. This algorithm is in fact very similar to the RAMBO algorithm [28] which is used to generate uniform phase space distributions.

Now it remains to work out the local density from Eq.(17). Some algebra leads to

$$\begin{aligned}
d^4\hat{p}_i\delta(\hat{p}_i^2 - m^2)d^4p_i\delta^4(p_i - xH_{\vec{b}}(\hat{p}_i)) &= \frac{1}{x^2}d^4p_i\delta(p_i^2 - x^2m^2) , \\
d^4\hat{k}_j\delta(\hat{k}_j^2)S(\hat{p}_1, \hat{p}_2, \hat{k}_j)d^4k_j\delta^4(k_j - xH_{\vec{b}}(\hat{k}_j)) &= d^4k_j\delta(k_j^2)S(p_1, p_2, k_j) , \\
\delta^3\left(\vec{b} + \frac{\vec{Q}}{\sqrt{Q^2}}\right) &= \frac{w^3}{\gamma}\delta^3(\vec{P}) , \\
\delta\left(x - \frac{w}{\sqrt{Q^2}}\right) &= \frac{w}{x}\delta(\sqrt{P^2} - w) , \\
\delta^3(\vec{\hat{p}}_1 + \vec{\hat{p}}_2) &= \frac{x^3R^0}{R^4}\delta^3\left(\vec{b} - \frac{\vec{R}}{\sqrt{R^2}}\right) , \\
\delta(\hat{p}_1^0 + \hat{p}_2^0 - w) &= \frac{x}{w}\delta\left(x - \frac{\sqrt{R^2}}{w}\right) , \tag{18}
\end{aligned}$$

where

$$R^\mu \equiv p_1^\mu + p_2^\mu , \quad P^\mu \equiv R^\mu + \sum_{j=1}^n k_j^\mu = xH_{\vec{b}}^\mu(Q) . \tag{19}$$

We can now proceed to collect the various terms in Eq.(17). Note that the delta functions which described the definition of $\vec{b}$ and x have now turned into the momentum conservation constraints on P^μ , while the original constraints on $\hat{p}_1^\mu + \hat{p}_2^\mu$ serve to eliminate the $\vec{b}$ and x from the integral: this implies that in this case (in contrast to that of the RAMBO algorithm) the transformation can actually be inverted, since for given p_i^μ and k_j^μ we can work out what the $\vec{b}$ and x were that gave this event. Moreover, note the nice feature that the part of the algorithm where the $\vec{b}$ and x are introduced and cancelled with the original momentum conservation constraints on $\hat{p}_1$ and $\hat{p}_2$ has exactly a unit Jacobian. We finally arrive at

$$\begin{aligned}
1 &= \frac{1}{A_f A_\gamma^n} \int \left[\prod_{i=1}^2 \frac{1}{(2\pi)^3} d^4p_i \delta(p_i^2 - x^2m^2) \right] \\
&\quad \left[\prod_{j=1}^n \frac{1}{(2\pi)^3} d^4k_j \delta(k_j^2) \right] \\
&\quad \delta^3(\vec{P}) \delta(P^0 - w) \\
&\quad \frac{w^4}{R^4} \left[\prod_{j=1}^n \psi(\hat{k}_j^0) S(p_1, p_2, k_j) \right] . \tag{20}
\end{aligned}$$

Apart from the mass fixup we are now finished, and the expression for the local density is given by the last line in Eq.(20):

$$F(p_1, p_2, k_1, \dots, k_n) = (A_f A_\gamma^n)^{-1} \frac{w^4}{R^4} \left[\prod_{j=1}^n \psi(\hat{k}_j^0) S(p_1, p_2, k_j) \right] , \quad (21)$$

where the $\hat{k}_j^0$ are given in terms of the actually generated momenta:

$$\hat{k}_j^0 = \frac{1}{x} H_{-b}^\mu(k_j) = \frac{w(R \cdot k_j)}{R^2} . \quad (22)$$

The Monte Carlo weight associated with a generated event, if for simplicity we take the exact matrix element to be unity ($A(p_1, p_2, k_1, \dots, k_n) = 1$), is then given by

$$W_{\text{MC}} = A_f A_\gamma^n \frac{R^4}{w^4} \left[\prod_{j=1}^n \psi(\hat{k}_j^0) S(p_1, p_2, k_j) \right]^{-1} . \quad (23)$$

As desired, this weight will compensate infrared and collinear peaks in the exact final-state radiation matrix element.

3.2 The regulator function

We now turn our attention to the choice of the regulator function $\psi(k)$. As mentioned above, it must vanish for $k \rightarrow 0$ and $k \rightarrow \infty$ sufficiently fast to make the photon spectrum integral in A_γ finite. At this point it is important to note the following: in the cross section as calculated by the Monte Carlo, there should not be any dependence on ϵ for small ϵ . However, this is not necessarily true for the *approximate* cross sections for the soft/virtual and hard bremsstrahlung contributions: a small ϵ dependence can be accepted there, as long as this can be compensated for by the Monte Carlo weights assigned to the events (see also appendix A).

To regulate the A_γ behaviour at small k the simplest choice is of course to cut the spectrum off below some value ϵ_1 . Note that there is a possible complication here, since the photon momenta are both boosted and scaled in the algorithm: one has to ensure that the photon phase space that one starts out with for the $\hat{k}_j^\mu$ includes completely the final desired photon phase space for the k_j^μ , otherwise a bias will develop since a part of the phase space of the k_j^μ will be underrepresented or even empty in the generated event sample. Fortunately the transformation from the $\hat{k}_j^\mu$ into the k_j^μ is invertible and we have the following inequality:

$$\begin{aligned} \hat{k}_j^0 &= \frac{1}{x} H_{-b}^0(k_j) = \frac{w}{R^2} (R^0 k_j^0 - \vec{R} \cdot \vec{k}_j) \\ &\geq k_j^0 \frac{w}{R^2} (R^0 - |\vec{R}|) = k_j^0 \frac{w}{R^0 + |\vec{R}|} \geq k_j^0 , \end{aligned} \quad (24)$$

where the last inequality results from the fact that (for fixed R^2) the maximum possible values for R^0 and $|\vec{R}|$ are $(w^2 + R^2)/2w$ and $(w^2 - R^2)/2w$, respectively. Therefore, the photon energy is always *degraded* by the algorithm and we can choose for ϵ_1 the value $w\epsilon/2$ that we already discussed. Note that the inequality can actually be saturated (when all photons are collinear with each other and back-to-back with the fermion pair) so that this value for ϵ_1 is essentially the strongest possible limit. We shall therefore consider regulator functions of the form

$$\psi(k) \propto \theta(k > \epsilon w/2) . \quad (25)$$

We now turn to the high-energy behaviour of ψ . The maximum allowable value of k is actually not $w/2$ as might be supposed naively: $\hat{k}_1^0 = w/2$ would (for $n = 1$) lead to $R^2 = w^2/2$ so that lower invariant masses of the fermion pair could not be reached. In order to obtain arbitrarily small R^2 values we need the possibility of arbitrarily large k_j^0 . One attractive-looking choice is

$$\psi(k) = \theta(k > \epsilon w/2) e^{-2k/w} , \quad (26)$$

so that

$$A_\gamma = \frac{2\alpha}{\pi} \log \left(\frac{1+\beta}{1-\beta} \right) E_1(\epsilon) \equiv \beta_f E_1(\epsilon) , \quad (27)$$

where $E_1(x)$ is the so-called exponential integral [29]:

$$E_1(x) \equiv \int_1^\infty \frac{e^{-xt}}{t} dt = \log \left(\frac{1}{x} \right) - \gamma_E - \sum_{n=1}^\infty \frac{(-x)^n}{n!n} , \quad (28)$$

and $\gamma_E = 0.577 \dots$ is Euler's constant. This leads to a well-defined exponentiation scheme since the contribution to the cross section from a final state with n final-state radiation photons will go approximately as $A_\gamma^n/n!$, and it is on the basis of these approximate values that we have to choose the multiplicity of final-state photons. Since for the choice (26) we have

$$\sum_{n=0}^\infty A_\gamma^n/n! = \exp(E_1(\epsilon)) \sim \left(\frac{1}{\epsilon} \right)^{\beta_f} e^{-\beta_f \gamma_E} \quad (29)$$

this poses no problem. That, however, the choice (26) is not acceptable becomes clear when we realize that in the Monte Carlo weight W_{MC} the fluctuations in the functions $S(p_1, p_2, k_j)$ are (hopefully) compensated by the behaviour of the exact cross section, but those in the factor

$$f(p_1, p_2, k_1, \dots, k_n) = \frac{R^4}{w^4 \prod_{j=1}^n \psi(\hat{k}_j^0)} \quad (30)$$

are not likewise compensated, so that the behaviour of f is important for the quality of the Monte Carlo algorithm. For the choice (26) we have

$$\prod_{j=1}^n \psi(\hat{k}_j^0) = \exp \left(- \sum_{j=1}^n 2\hat{k}_j^0/w \right) = \exp \left(2 - 2 \frac{wR^0}{R^2} \right) , \quad (31)$$

so that

$$f(p_1, p_2, k_1, \dots, k_n) = \frac{R^4}{w^4} \exp \left(\frac{2wR^0}{R^2} - 2 \right) . \quad (32)$$

Hence for small values of R^2 the weight factor f explodes: whereas $f \sim 1$ for $R^2 \sim w^2$, for $\sqrt{R^2}/w = 0.2$, which is not extremely small, f varies between 4.77 and 4.24×10^7 ! So unless strict cuts on R^2 are intended this choice of ψ is inadequate. Physically, it means that the exponential suppression of the photon spectrum is too severe: it leads to an underrepresentation of events with high Q^2 and consequently low R^2 , for which the algorithm must make up by assigning to such events a very large weight.

A possible choice with a harder photon spectrum would be, for instance,

$$\psi(k) = \theta(k > \epsilon w/2) \frac{w}{w + 2k} , \quad (33)$$

leading to

$$\begin{aligned} A_\gamma &= \beta_f \log \left(\frac{1 + \epsilon}{\epsilon} \right) , \\ \sum_{n=0}^{\infty} \frac{1}{n!} A_\gamma^n &= \left(\frac{1}{\epsilon} \right)^{\beta_f} (1 + \mathcal{O}(\epsilon)) . \end{aligned} \quad (34)$$

We would find, however, for f :

$$f(p_1, p_2, k_1, \dots, k_n) = \frac{R^{4-2n}}{w^4} \prod_{j=0}^n (R + k_j)^2 . \quad (35)$$

Although the weight no longer diverges exponentially it still blows up for small R^2 . Consequently we need an even harder spectrum for the photons. We could therefore consider

$$\psi(k) = \theta(k > \epsilon w/2) \left(\frac{w}{w + 2k} \right)^{2/n} , \quad (36)$$

which would imply

$$f(p_1, p_2, k_1, \dots, k_n) = \frac{1}{w^4} \left[\prod_{j=1}^n (R + k_j)^2 \right]^{2/n} , \quad (37)$$

with no pole left as $R^2 \rightarrow 0$. For A_γ the choice (36) yields

$$\begin{aligned} A_\gamma &= \beta_f \int_\epsilon^\infty dx \frac{1}{x(1+x)^{2/n}} \\ &= \frac{n}{2\epsilon^{2/n}} F\left(\frac{2}{n}, \frac{2}{n}; \frac{n+2}{n}; -\frac{1}{\epsilon}\right) \end{aligned} \quad (38)$$

$$= \left[\frac{1}{4^n}, 1\right] \beta_f \left(\log\left(\frac{1}{\epsilon}\right) + \frac{2}{n}\right) . \quad (39)$$

Here F is the hypergeometric function [29]: it is easily shown that A_γ is bounded by the expression in Eq.(39), where the lower bound comes from taking the 4^{-n} and the upper bound from taking the 1 in the square brackets. The total multiplicity sum therefore reads

$$\sum_{n=0}^\infty \frac{1}{n!} A_\gamma^n = \left[\frac{1}{4}, 1\right] \sum_{n=0}^\infty \frac{\beta_f^n}{n!} \left(\log\left(\frac{1}{\epsilon}\right) + \frac{n}{2}\right)^n . \quad (40)$$

For reasonable values of ϵ this sum is actually quite well-behaved: it equals $[1/4, 1]\epsilon^{-\beta_f}$ to within a factor of 2 for all ϵ between 1 and 10^{-30} . Therefore this regulator function is acceptable for Monte Carlo simulations. Note, however, that for large multiplicities, $n \gg \log(1/\epsilon)$, we have

$$\frac{\beta_f^n}{n!} A_\gamma^n \sim \left(\frac{e\beta_f}{2}\right)^n \frac{1}{\sqrt{2\pi n}} \left(\frac{1}{\epsilon}\right)^2 , \quad (41)$$

so that for these high multiplicities the Monte Carlo weights will come out very low: after all, the *real* ϵ dependence of the contribution from n photons is $\log(1/\epsilon)^n$ rather than $1/\epsilon^2$.

An even nicer behaviour of the algorithm can in principle be obtained by introducing not a regulator function $\psi(\hat{k}_j^0)$ for each bremsstrahlung photon separately, but a single function $\psi(\hat{k}_1^0 + \hat{k}_2^0 + \dots \hat{k}_n^0)$ for the whole set of photons (for given n). Taking the choice

$$\psi(\hat{k}_1^0 + \hat{k}_2^0 + \dots \hat{k}_n^0) = \left[\prod_{j=1}^n \theta(\hat{k}_j^0 > w\epsilon/2) \right] \left(\frac{w}{w + \sum_{m=1}^n \hat{k}_m^0} \right)^2 , \quad (42)$$

we arrive at the excellent result

$$f(p_1, p_2, k_1, \dots, k_n) = \left(\frac{R^0}{w}\right)^2 ; \quad (43)$$

this is about the nicest Monte Carlo weight one could envisage after a strictly constant weight. The choice (42) implies that the photons cannot be generated independently: moreover A_γ^n must now be replaced by

$$A_\gamma^n \rightarrow K_n(\epsilon) \equiv \int_{\epsilon/2}^\infty \left[\prod_{i=1}^n \frac{dx_i}{x_i} \right] \frac{1}{\left(1 + \sum_{j=1}^n x_j\right)^2} . \quad (44)$$

This ugly-looking expression can be made more amenable by Laplace transformation:

$$\begin{aligned}
K_n(\epsilon) &= \int_{\epsilon/2}^{\infty} \prod_{i=1}^n dx_i \int_0^{\infty} \prod_{i=1}^n dz_i \int_0^{\infty} dz \, z e^{-z(1+\sum_{m=1}^n x_m)} e^{-\sum_{j=1}^n x_j z_j} \\
&= \int_0^{\infty} dz \, z e^{-z} \int_0^{\infty} \prod_{i=1}^n \left[dz_i \frac{e^{-(z+z_i)\epsilon/2}}{z+z_i} \right] \\
&= \int_0^{\infty} dz \, z e^{-z} E_1(z\epsilon/2)^n .
\end{aligned} \tag{45}$$

Although we do not know a closed form for the result of Eq.(45), it can be readily evaluated for any given ϵ and n ; moreover, what interests us here are the convergence properties of the total multiplicity sum:

$$\begin{aligned}
\sum_{n=0}^{\infty} \frac{\beta_f^n}{n!} K_n(\epsilon) &= \int_0^{\infty} dz \, z e^{-z} \exp(\beta_f E_1(z\epsilon/2)) \\
&= \left(\frac{2}{\epsilon}\right)^{\beta_f} e^{-\beta_f \gamma_E} (\Gamma(2 - \beta_f) + \mathcal{O}(\epsilon)) ,
\end{aligned} \tag{46}$$

where we have used the explicit representation (28) for the exponential integral $E_1(z\epsilon/2)$. We see that the multiplicity sum behaves also in an attractive way, going with the correct power of $1/\epsilon$. It might be observed that also in this case the integral $K_n(\epsilon)$ will behave as $1/\epsilon^2$ for asymptotically large n : however, in contrast to the case of Eq.(36), this only happens for $n \gg \epsilon^{-1}$ rather than for $n \gg \log(1/\epsilon)$: such astronomically large multiplicities are well outside the range that we want to study by Monte Carlo, and moreover the soft-photon approximation breaks down at that point, since $\sum_j k_j^0$ is then no longer negligible with respect to w .

So far we have considered choices for ψ that allow one to generate all possible kinematical configurations for the final state, including arbitrarily small R^2 . The choices (36) and (42) appear to be appropriate to this end: however, the corresponding photon energy spectra are not trivial to generate (see appendix B). Moreover, in many if not all applications one will want to impose an invariant-mass cut on the final fermion momenta. If, as before, a lower mass-squared limit of s_0 is assumed, this leads to an upper limit on the values of $\hat{k}_j^0$ that must be generated. In the single-bremsstrahlung case this limit is given by

$$\hat{k}_1^0 \leq \frac{w}{2} \left(\frac{w^2}{s_0} - 1 \right) ; \tag{47}$$

for multiple bremsstrahlung the actual limit is of course more complicated, involving the angles of the photons as well as their energies, but as the addition of a photon decreases the fermion pair mass even further, the resulting upper limits on the photon

energies are always bounded from above by the limit of Eq.(47). We therefore take this value as the general upper limit for all n , so that

$$\psi(\hat{k}_j^0) = \theta \left(\frac{\epsilon w}{2} < \hat{k}_j^0 < \frac{w}{2} \left(\frac{w^2}{s_0} - 1 \right) \right) , \quad (48)$$

leading to

$$\begin{aligned} f(p_1, p_2, k_1, \dots, k_n) &= \frac{R^4}{w^4} , \\ A_\gamma &= \beta_f \log \left(\frac{1}{\epsilon} \left(\frac{w^2}{s_0} - 1 \right) \right) , \\ \sum_{n=0}^{\infty} \frac{A_\gamma^n}{n!} &= \left(\frac{1}{\epsilon} \left(\frac{w^2}{s_0} - 1 \right) \right)^{\beta_f} . \end{aligned} \quad (49)$$

These photon spectra are very simple to generate, and as we see the behaviour of the multiplicity sum is quite acceptable as long as s_0 is not unrealistically small. We shall therefore adopt the choice (48) as the preferred one. This brings us to the last step in the construction of the Monte Carlo algorithm: we have to make sure to stay within the desired phase space, in which we now must have $k_j^0 > w\epsilon/2$ as well as $R^2 > s_0$. The Monte Carlo weight as indicated in Eq.(23) is therefore

$$W_{\text{MC}} = \frac{R^4}{w^4} \theta(R^2 \geq s_0) \prod_{j=1}^n \left[\theta \left(k_j^0 \geq \frac{\epsilon w}{2} \right) S(p_1, p_2, k_j)^{-1} \right] . \quad (50)$$

This finishes our discussion of the main part of the generating algorithm.

3.3 The mass fixup transformation

We now turn our attention to the problem of determining the (small) change we have to make in the generated momenta p_i^μ, k_j^μ so that the masses of the fermions become equal to m rather than xm . This procedure has already been described in [28] but we give it here for completeness. It consists of scaling the three-momenta of all particles by a factor ξ , and assigning the appropriate energy component to all momenta so that each particle is on the correct mass shell. The value of ξ is then fixed by the requirement that the sum of the energies equals w . For the sake of generality we shall start with a set of momenta p_i^μ ($i = 1, \dots, N$) with masses μ_i , which we want to transform into a set of momenta q_i^μ with masses m_i : later we shall specify to our particular case. The phase space for the p_i^μ (leaving out irrelevant factors of 2π) is given by

$$W = \int \left[\prod_{i=1}^N d^4 p_i \delta(p_i^2 - \mu_i^2) \right] \delta^3 \left(\sum_{i=1}^N \vec{p}_i \right) \delta \left(\sum_{i=1}^N p_i^0 - w \right) . \quad (51)$$

The mass fixup transformation is given by

$$\begin{aligned}\vec{p}_i &\rightarrow \vec{q}_i \equiv \xi \vec{p}_i, \\ p_i^0 &\rightarrow q_i^0 \equiv \sqrt{\xi^2 \vec{p}_i^2 + m_i^2},\end{aligned}\quad (52)$$

where ξ solves

$$\sum_{i=1}^N \sqrt{\xi^2 \vec{p}_i^2 + m_i^2} - w = 0; \quad (53)$$

this solution we denote by $U(\vec{p}, m)$ to signify that ξ depends only on these quantities (and not, for instance, on the μ_i). There is no general closed form for $U(\vec{p}, m)$ but it can easily be computed numerically using the Newton-Raphson method. The algorithm leading from the p_i^μ to the q_i^μ can now be written as

$$\begin{aligned}W = \int & \left[\prod_{i=1}^N \frac{d^3 \vec{p}_i}{2p_i^0} \right] \delta^3 \left(\sum_{i=1}^N \vec{p}_i \right) \delta \left(\sum_{i=1}^N p_i^0 - w \right) d\xi \delta(\xi - U(\vec{p}, m)) \\ & \left[\prod_{i=1}^N d^4 q_i \delta^3(\vec{q}_i - \xi \vec{p}_i) \delta \left(q_i^0 - \sqrt{\xi^2 \vec{p}_i^2 + m_i^2} \right) \right].\end{aligned}\quad (54)$$

Proceeding as before we can rewrite this as

$$\begin{aligned}W = \int & \left[\prod_{i=1}^N \frac{1}{\xi^3} \frac{q_i^0}{p_i^0} d^4 q_i \delta(q_i^2 - m_i^2) \right] \xi^3 \delta \left(\sum_{i=1}^N \vec{q}_i \right) \\ & d\xi \delta(\xi - U(\vec{p}, m)) \delta \left(\sum_{i=1}^N p_i^0 - w \right).\end{aligned}\quad (55)$$

We can interchange the rôles of the two Dirac functions in the last line by writing

$$\begin{aligned}\delta(\xi - U(\vec{p}, m)) &= \frac{1}{x} \left(w - \sum_{i=1}^N \frac{m_i^2}{q_i^0} \right) \delta \left(\sum_{i=1}^N q_i^0 - w \right), \\ \delta \left(\sum_{i=1}^N p_i^0 - w \right) &= \left[\frac{1}{\xi} \left(w - \sum_{i=1}^N \frac{\mu_i^2}{p_i^0} \right) \right]^{-1} \delta \left(\xi - \frac{1}{U(\vec{q}, \mu)} \right).\end{aligned}\quad (56)$$

We end up with a form for the phase space in which the local density for this part of the algorithm is explicit:

$$\begin{aligned}W = \int & \left[\prod_{i=1}^N d^4 q_i \delta(q_i^2 - m_i^2) \right] \delta^3 \left(\sum_{i=1}^N \vec{q}_i \right) \delta \left(\sum_{i=1}^N q_i^0 - w \right) \\ & \xi^{3-3N} \left[\prod_{i=1}^N \frac{q_i^0}{p_i^0} \right] \left[w - \sum_{i=1}^N \frac{m_i^2}{q_i^0} \right] \left[w - \sum_{i=1}^N \frac{\mu_i^2}{p_i^0} \right]^{-1}.\end{aligned}\quad (57)$$

The Monte Carlo weight associated with this step is therefore given by

$$W_{\text{MC}}^{(2)} = \xi^{3N-3} \left[\prod_{i=1}^N \frac{p_i^0}{q_i^0} \right] \frac{w - \sum_{i=1}^N \frac{\mu_i^2}{p_i^0}}{w - \sum_{i=1}^N \frac{m^2}{q_i^0}} , \quad (58)$$

which can be added as an additional factor to the weight W_{MC} discussed before.

We can now specialize to the case under consideration: the $p_{1,2}^\mu$ are the generated fermion momenta with mass xm , and the $p_{3,\dots,N}^\mu$ are the massless photon momenta k_j^μ . After the mass fixup transform, the $q_{1,2}^\mu$ must become the fermion momenta with mass m , and the $q_{3,\dots,N}^\mu$ the resulting photon momenta. The equation leading to $\xi = U(\vec{p}, m)$ can then be written as

$$H(\xi) \equiv \sqrt{\xi^2 \vec{p}_1^2 + m^2} + \sqrt{\xi^2 \vec{p}_2^2 + m^2} - \xi p_1^0 - \xi p_2^0 - w(1 - \xi) = 0 , \quad (59)$$

and the additional Monte Carlo weight takes the form

$$W_{\text{MC}}^{(2)} = \xi^{2n+3} \left(\frac{p_1^0 p_2^0}{q_1^0 q_2^0} \right) \left[w - \frac{x^2 m^2}{p_1^0} - \frac{x^2 m^2}{p_2^0} \right] \left[w - \frac{m^2}{q_1^0} - \frac{m^2}{q_2^0} \right]^{-1} . \quad (60)$$

Using these results the mass fixup transform can easily be performed, and the corresponding weight multiplied into the total Monte Carlo weight. It is important to note that, since x is always smaller than 1, ξ is also always smaller than 1, and again the photon energies are always degraded by the algorithm: this ensures once more that we are allowed to use the lower limit $\epsilon w/2$ in the generation of the photon momenta. Another point of interest is that since in the majority of the events the photon energies are quite low, x and also ξ tend to be close to 1. In addition, when $m = 0$ no mass fixup transform would be needed, so in that case $\xi = 1$ and $W_{\text{MC}}^{(2)} = 1$. When the fermion mass is quite small (as in the case of Bhabha scattering) the relativistic limit is therefore relevant. In this limit, Eq.(59) reduces to

$$\xi = 1 - (1 - x^2) \frac{m^2}{2w} \left(\frac{1}{p_1^0} + \frac{1}{p_2^0} \right) + \mathcal{O}(m^4) \quad (61)$$

and the weight factor then reads

$$W_{\text{MC}}^{(2)} = \xi^{2n-1} \left(1 - (1 - x^2) \frac{m^2}{2} \left(\frac{1}{(p_1^0)^2} + \frac{1}{(p_2^0)^2} \right) \right) + \mathcal{O}(m^4) . \quad (62)$$

Another useful observation is that in the present case, with two massive particles and the rest massless, we can rewrite Eq.(53) as

$$q_1^0 - \xi p_1^0 + q_2^0 - \xi p_2^0 - w(1 - \xi) = 0 : \quad (63)$$

by virtue of $q_i^0 < \xi |\vec{p}_i| + m$ and $p_i^0 > |\vec{p}_i|$ this implies

$$1 - 2m/w \leq \xi \leq 1 . \quad (64)$$

This is a loose bound (however already quite strict for small m): better bounds can be derived but they depend on q_1^0, q_2^0 , and x .

One important remark is in order here. Since the mass fixup transform affects the momenta of the two fermions their invariant mass is also changed. Introduce the following notation: $P = \sum_{i=0} N p_i$, $\tilde{R} = p_1 + p_2$, $R = q_1 + q_2$, $\tilde{K} = P - \tilde{R}$, $K = P - R$, $\tilde{s}' = \tilde{R}^2$ and $s' = R^2$. We then have for Δ_s , the change in the fermion invariant mass squared due to the mass fixup transformation:

$$\begin{aligned} \Delta_s &\equiv s' - \tilde{s}' = (P - K)^2 - (P - \tilde{K})^2 \\ &= \xi^2 \tilde{K}^2 - 2\xi P \cdot \tilde{K} + (2P \cdot \tilde{K} + \tilde{K}^2) , \end{aligned} \quad (65)$$

which is positive for all $\xi < 1$. An upper bound on Δ_s is given by the following estimate:

$$\begin{aligned} \Delta_s &= 2P \cdot K \left(\frac{1}{\xi} - 1 \right) - K^2 \left(\frac{1}{\xi^2} - 1 \right) \\ &= \frac{1 - \xi}{\xi} \left(2P \cdot K - \left(\frac{1}{\xi} + 1 \right) K^2 \right) \\ &< 2 \frac{1 - \xi}{\xi} (P \cdot K - K^2) \\ &= \frac{1 - \xi}{\xi} (w^2 - s' - K^2) \\ &= \frac{1 - \xi}{\xi} (w^2 - s') . \end{aligned} \quad (66)$$

Note that this estimate is quite sharp since we can have $K^2 = 0$ (for instance if $n = 1$) and ξ is typically close to 1. Putting in the lower limit $\xi = 1 - 2m/w$ we find

$$s' - \tilde{s}' < \frac{2m}{w - 2m} (w^2 - s') . \quad (67)$$

Applied naively for a too low value of s_0 the transformation may underpopulate the region of low fermion pair mass (*cf.* the discussion on the transformation of the photon energy lower bound in the previous section) since of course $\tilde{s}'$ cannot be less than 0. This implies two things: in the first place, the algorithm should be run with as a lower cutoff on the fermion pair mass not s_0 but $(ws_0 - 2mw^2)/(w - 2m)$; and in the second place, s_0 should not be smaller than $2mw$ in order to avoid the abovementioned underpopulation. For electrons at LEP energies this lower bound is about $s_0 > 0.3 \text{ GeV}$; for τ leptons it comes to about 18 GeV. The lower bound on s_0

could in principle be improved (that is, lowered) by improving (that is, raising) the lower bound on ξ for given s' , since typically $\xi = 1 - \mathcal{O}(m^2)$ and not $1 - \mathcal{O}(m)$.

Before finishing we should note that also the mass fixup transform is invertible, in the sense that we can, for given q_i^μ , determine x and ξ . This is due to the fact that only two of the momenta are massive, and can most easily be done by introducing $\eta = 1 - \xi$, $\sigma = 1 - x^2 \xi^2$, and $u_i(\sigma) \equiv \xi p_i^0 = \sqrt{q_i^{02} - \sigma m^2}$. The equations defining ξ (Eq.(59)) and x can then be written as follows:

$$\begin{aligned} q_1^0 + q_2^0 - u_1(\sigma) - u_2(\sigma) - w\eta &= 0 , \\ (p_1 + p_2)^2 &= x^2 w^2 . \end{aligned} \quad (68)$$

Some algebra leads then to

$$\begin{aligned} \sigma &= \frac{1}{A} (B - \sqrt{B^2 - AC}) , \\ A &= w^2(w^2 - 4m^2) , \\ B &= (w^2 - 2m^2)(w^2 - s' + 2q_1^0 q_2^0) - 2m^2(q_1^{02} + q_2^{02}) , \\ C &= (w^2 - s' + 2q_1^0 q_2^0)^2 - (2q_1^0 q_2^0)^2 , \\ \eta &= \frac{1}{w} ((q_1^0 + q_2^0 - u_1(\sigma) + u_2(\sigma)) . \end{aligned} \quad (69)$$

In this way we can, even after the mass fixup transform, calculate the local density of the algorithm.

4 Conclusions

We have argued that for high-precision phenomenology final-state radiative corrections and the interference between initial- and final-state radiation will be relevant. We have seen that the YFS approach to radiative corrections can lead naturally to a Monte Carlo implementation. We have reviewed possible algorithms to generate multiphoton final states. Obviously a high-precision Monte Carlo program will deal with much more than only final-state radiation, although the above considerations have to be implemented in it. An extensive description of such a Monte Carlo, the small-angle Bhabha scattering generator BLUES, will be presented elsewhere [27].

Appendix A: The soft-photon approximation in Monte Carlo

In generating events, the photon multidifferential cross section is given by Eq.(11) in which, as we have said, some infrared divergent terms are left out: in the usual soft

photon approximation, Eq.(11) reads

$$S'(\hat{p}_1, \hat{p}_2, \hat{k}_j) = e^2 \left[\frac{(\hat{p}_1 + \hat{p}_2)^2}{(\hat{p}_1 \cdot \hat{k}_j)(\hat{p}_2 \cdot \hat{k}_j)} - \frac{m^2}{(\hat{p}_1 \cdot \hat{k}_j)^2} - \frac{m^2}{(\hat{p}_2 \cdot \hat{k}_j)^2} \right] . \quad (70)$$

Integrating Eq.(70) over photon energies from $\epsilon w/2$ up to, say, $w/2$, we obtain

$$\int dk S'(\hat{p}_1, \hat{p}_2, \hat{k}_j) = \frac{2\alpha}{\pi} \left[\log \left(\frac{2\hat{p}_1 \cdot \hat{p}_2}{m^2} \right) - 1 \right] \log \left(\frac{1}{\epsilon} \right) , \quad (71)$$

whereas Eq.(11) leads to the above result without the (-1) term, hence mismatching the corresponding term from virtual photons: the approximation $S(\hat{p}_1, \hat{p}_2, \hat{k}_j)$ overestimates the hard bremsstrahlung contribution. Hence a mismatch develops in the cancellation between the infrared terms in the virtual and in the real photon contributions: after exponentiation this causes the *approximate* cross section (on which the Monte Carlo is based) to depend, after all, on ϵ as $\exp(2\alpha \log(1/\epsilon)/\pi)$. This is of course trivially made up for by the assignment of a correspondingly lower Monte Carlo weight for hard photon events, and the *computed* cross section will be ϵ -independent, but problems might arise if those weights come out dramatically small. From the above we see that the overestimate of the hard photon cross section will be about a factor of 2 only for $\epsilon \sim \exp(-2\pi \log(2)/\alpha) \sim 10^{-64}$.

The motivation for using Eq.(11) rather than Eq.(70) is the fact that $S'(\hat{p}_1, \hat{p}_2, \hat{k}_j)$ becomes exactly zero if $\vec{k}_j$ is parallel to $\vec{p}_{1,2}$ in their centre-of-mass frame. Since in the exact matrix element this exact cancellation holds for zero-energy but not for finite-energy photons, the use of Eq.(70) would imply using an approximant which is zero in places where the actual function is non-zero, leading to infinitely large weights. As an alternative to Eq.(11) one might consider soft-photon-like approximants that do go to zero in the collinear case only for $\hat{k}_j^0 \rightarrow 0$, but such an approximant is clearly not homogeneous in $\hat{k}_j^0$ and hence the nice conformal properties of Eq.(11) would be spoiled. The above argument, which is included here for completeness, has earlier been given in [30, 31].

Appendix B: Algorithms for the photon energies

In this appendix we give explicit algorithms to generate the hatted photon energies $\hat{k}_j^0$ ($j = 1, \dots, n$) discussed in section 3. We here list the various regulator functions and the corresponding algorithms in pseudocode. As before, r_i denote pseudorandom numbers equidistributed between 0 and 1.

1. **Exponentially damped photon spectrum (Eq.(26)).** The distribution to be generated is

$$\psi(x) = \frac{e^{-x}}{x} \theta(x \geq a) ; \quad (72)$$

the algorithm is

$$\begin{aligned}
&\textbf{repeat} \\
&\quad x \leftarrow \left[\left(\frac{1+a}{a} \right)^{r_1} - 1 \right]^{-1} \\
&\quad w \leftarrow \frac{1+x}{1+a} e^{a-x} \\
&\textbf{until } w > r_2 .
\end{aligned} \tag{73}$$

This is in fact the generation of the next distribution, $\theta(x \geq a)/(x(1+x))$, followed by a rejection step to bring the damping behaviour down from $(1+x)^{-1}$ to e^{-x} . The efficiency, that is the average number of trials in the repeat loop, is $e^a E_1(a)/(1+a) \log(1+1/a)$, which is around 40% for $a = 1$ and goes to 100% as $a \rightarrow 0$.

2. **‘Linearly’ damped photon spectrum (Eq.(33)).** The distribution to be generated is indicated by the previous one:

$$\psi(x) = \frac{1}{x(1+x)} \theta(x \geq a) ; \tag{74}$$

the algorithm is

$$x \leftarrow \left[\left(\frac{1+a}{a} \right)^{r_1} - 1 \right]^{-1} . \tag{75}$$

This is of course 100% efficient.

3. **‘Power- ν ’ damped photon spectrum (Eq.(36)).** The distribution to be generated is

$$\psi(x) = \frac{1}{x(1+x)^\nu} \theta(x \geq a) , \tag{76}$$

with $\nu = 2/n$. For small ν the simplest algorithm appears to be

$$\begin{aligned}
&P \leftarrow \frac{\log(1/a)}{\log(1/a) + 1/\nu} \\
&\textbf{repeat} \\
&\quad \textbf{if } r_1 < P \textbf{ then} \\
&\quad \quad x \leftarrow a^{r_2} \\
&\quad \quad w \leftarrow (1+x)^{-\nu} \\
&\quad \textbf{else} \\
&\quad \quad x \leftarrow r_3^{-1/\nu} \\
&\quad \quad w \leftarrow \left(\frac{x}{1+x} \right)^\nu \\
&\quad \textbf{end if} \\
&\textbf{until } w > r_3 .
\end{aligned} \tag{77}$$

which has an efficiency bounded by $[2^{-\nu}, 1]$. For $n = 2$ ($\nu = 1$) the algorithm of the previous case is of course preferable, while for $n = 1$ ($\nu = 2$) it is better to use also that algorithm, but with an additional weighting-rejection step:

$$\begin{aligned}
&\textbf{repeat} \\
&\quad x \leftarrow \left[\left(\frac{1+a}{a} \right)^{r_1} - 1 \right]^{-1} \\
&\quad w \leftarrow \frac{1+a}{1+x} \\
&\textbf{until } w > r_2 .
\end{aligned} \tag{78}$$

In this case the efficiency is $1 - 1/[(1+a)\log((1+a)/a)]$ which goes to 100% as $a \rightarrow 0$.

4. **Combined multiphoton spectrum (Eq.(42)).** The n -dimensional distribution to be generated is

$$\psi(x_1, \dots, x_n) = \frac{\prod_{i=1}^n \theta(x_i \geq a)}{(1 + \sum_{i=1}^n x_i)^2 \prod_{i=1}^n x_i} \tag{79}$$

This distribution is quite nontrivial to generate, but we can attack it by considering Eq.(45). It appears that we can generate a z according to the last line in Eq.(45): having z , the z_i are then distributed according to $\exp(-a(z_i + z))/(z_i + z)$; once the z_i are obtained, the x_i are generated from the distribution $\exp(-x_i(z_i + z))$. The inclusive distribution for the z from Eq.(45) is by itself also not easily generated, but we must realize that also n is a variable that has to be generated. Therefore we may also sum over the n , generate the z according to Eq.(46), and then pick n from the Poisson distribution with mean value $\beta_f E_1(az)$. Explicitly, this nest of algorithms reads:

(a) **to obtain z :**

$$\begin{aligned}
&\textbf{repeat} \\
&\quad z \leftarrow r_1/r_2 \\
&\quad \textbf{if } z > 1 \textbf{ then} \\
&\quad \quad y \leftarrow z^2 \\
&\quad \textbf{else} \\
&\quad \quad y \leftarrow 1 \\
&\quad \textbf{end if} \\
&\quad w \leftarrow \frac{yz \exp(-z + \beta_f E_1(az))}{\exp(\beta_f(E_1(1) - \log(a)))} \\
&\textbf{until } w > r_3 .
\end{aligned} \tag{80}$$

(b) to obtain n :

$$\begin{aligned}
& \nu \leftarrow E_1(az) \\
& n \leftarrow -1 \\
& P \leftarrow 0 \\
& s \leftarrow r_4 \\
& \textbf{repeat} \\
& \quad n \leftarrow n + 1 \\
& \quad P \leftarrow P + \frac{e^{-\nu} \nu^n}{n!} \\
& \textbf{until } P > s .
\end{aligned} \tag{81}$$

(c) to obtain the z_i :

$$\begin{aligned}
& \textbf{do } i = 1, n \\
& \quad \textbf{obtain } t_1 \textbf{ from } e^{-t}/t\theta(t \geq az) \textbf{ as in Eq.(73)} \\
& \quad z_i \leftarrow -z + t_i/a \\
& \textbf{end do } .
\end{aligned} \tag{82}$$

(d) to obtain the x_i :

$$\begin{aligned}
& \textbf{do } i = 1, n \\
& \quad x_i \leftarrow a - \frac{\log(r_5)}{z_i + z} \\
& \textbf{end do } .
\end{aligned} \tag{83}$$

5. Cut-off photon spectrum (Eq.(48)). The spectrum reads

$$\psi(x) = \frac{1}{x} \theta(a \leq x \leq b) , \tag{84}$$

and the algorithm is extremely simple and 100% efficient:

$$x \leftarrow a \left(\frac{b}{a} \right)^{r_1} . \tag{85}$$

References

- [1] *Z physics at LEP1*, eds. G. Altarelli, R. Kleiss and C. Verzegnassi, CERN Yellow Report CERN 89-09 (3 vols.).
- [2] F. Dydak, preprint CERN-PPE/91-15.

- [3] ‘Z line shape’, D. Bardin *et al.*, in [1].
- [4] O. Nicrosini and L. Trentadue, Phys. Lett. **B196**(1987)551.
- [5] F.A. Berends, G. Burgers and W. van Neerven, Phys. Lett. **B177** (1986) 191.
- [6] D. Bardin *et al.*, Nucl. Phys. **B351**(1991)1, Phys. Lett. **B255**(1991)290, and references quoted therein.
- [7] B.W. Lynn *et al.*, Nucl. Phys. **B321**(1989)83.
- [8] G. Montagna, O. Nicrosini and L. Trentadue, Phys. Lett. **PB231**(1990)492.
- [9] ‘Forward-backward asymmetries’, D. Bardin *et al.*, in [1].
- [10] T. Riemann and Z. Wąs, Mod. Phys. Lett. **A4**(1989)2487.
- [11] ‘Monte Carlos for electroweak physics’, D. Bardin *et al.*, in [1].
- [12] J.P. Alexander *et al.*, Phys Rev. **D37**(1988)56.
- [13] W. de Boer, unpublished: see also [11].
- [14] F.A. Berends, R. Kleiss and S. Jadach, Nucl. Phys. **B202**(1982)63.
- [15] F.A. Berends, R. Kleiss and W. Hollik, unpublished: see also [16]
- [16] F.A. Berends, R. Kleiss and W. Hollik, Nucl. Phys. **B304**(1988)712.
- [17] F.A. Berends and R. Kleiss, Nucl. Phys. **B228**(1983)537.
- [18] G. Bonvicini and L. Trentadue, Nucl. Phys. **B323**(1989)253.
- [19] J.-E. Campagne and R. Zitoun, Z.f. Phys. **C43**(1988)469; J.-E. Campagne, Ph.D. thesis, LPN-HEP 89.02.
- [20] R. Kleiss, Nucl. Phys. **B347**(1990)67.
- [21] S. Jadach and B.F.L. Ward, Phys. Rev. **D38**(1988)2897; E: **D39**(1989)1471.
- [22] D.R. Yennie, S. Frautschi and H. Suura, Ann. Phys. **13**(1961)379.
- [23] KORALZ in [11] and CERN-TH.5994/91 (1991).
- [24] S. Jadach and B.F.L. Ward, Phys. Rev. **D40**(1989)3582.
- [25] S. Jadach and Z. Wąs, Phys. Rev. **D41**(1990)1425.

- [26] T.D. Lee and M. Nauenberg, Phys. Rev. **133**(1964)1549; T. Kinoshita, J. Math. Phys. **3**(1962)650.
- [27] R. Kleiss, in preparation.
- [28] S.D. Ellis, R. Kleiss and W.J. Stirling, Comp. Phys. Comm. **40**(1986)359.
- [29] M. Abramowitz and I.A. Stegun, *Handbook of mathematical functions* (Dover, New York 1972).
- [30] S. Jadach, [21] and private communication.
- [31] S. van der Marck, Ph. D. thesis, Leiden 1991.