CERN-TH. 6230c/91

Monte Carlo program BHLUMI 2.01 for Bhabha scattering at low angles with Yennie-Frautschi-Suura exponentiation

S. Jadach

CERN, Theory Division, Geneva 23, Switzerland,

and

Institute Physics, Jagellonian University, Kraków, ul. Reymonta 4, Poland

E. Richter-Wąs

Chair of Computer Science, Jagellonian University,

Kraków, ul. Reymonta 4, Poland

B.F.L. Ward†

Department of Physics and Astronomy,

The University of Tennessee, Knoxville, Tennessee 37996-1200

and

SLAC, Stanford University, Stanford, California 94309

Z. Wąs

Institute of Nuclear Physics, Kraków, ul. Kawiory 26a, Poland

Abstract

The Monte Carlo program for small-angle Bhabha scattering with an overall precision of 0.25% is presented. The QED calculation at this precision level is of vital importance for luminosity measurement at LEP/SLC experiments. BHLUMI is a stand-alone Monte Carlo event generator with three sub-generators: BHLUM2, LUMLOG and OLDBIS. The first of them is based on Yennie-Frautschi-Suura $\mathcal{O}(\alpha)$ exponentiation and the two other ones serve, essentially, to cross-check results from the first one. LUMLOG represents pure leading logarithmic calculation up to $\mathcal{O}(L^3\alpha^3)$ and OLDBIS represents ordinary (non-exponentiated) $\mathcal{O}(\alpha)$ calculation. The important advantage of LUMLOG and OLDBIS is that they feature very high *technical* precision of 0.02%. All three sub-generators form a very powerful tool box of programs for precise calculations of QED corrections to the luminosity measurements in present $e^{\pm}$ annihilation experiments.

Published in Computer Physics Communications: **70** *(1992) 305.*

† Work supported in part by the US DOE contracts DE-AS05-76ER03956 and DE-AC03-76SF00515 and by TNRLC grant RCFY9101

CERN-TH. 6230/91
September, 1991

PROGRAM SUMMARY

Title of the program: BHLUMI version 2.01
Computer: IBM 3090, APOLLO DN-10000; Installation: CERN, SLAC
Operating system: VM/CMS
Programming language used: FORTRAN 77
High speed storage required: 45000 words
No. of bits in a word: 32
Peripherals used: Line printer
No. of cards in combined program and test deck: about 6600
Keywords: Radiative corrections, Monte Carlo simulation, Bhabha scattering, bremsstrahlung, Quantum Electrodynamics (QED), electroweak theory, structure functions.
Nature of physical problem: Small-angle Bhabha scattering process is used in all electron-positron colliders to calculate machine luminosity. This process is subject to QED radiative corrections which has to be known for arbitrary cut-offs and/or acceptance with precision at least factor three better then pure experimental precision. It means that the level of 0.2% should be reached. The realistic simulation should include multiple emission of the bremsstrahlung photons.
Method of solution: The Monte Carlo simulation of the small-angle process is an ideal solution. It provides integrated cross-section for arbitrary cuts. Direct simulation of the final-state electrons and photos is precisely what is needed for detector simulation purpose.
Restrictions on the complexity of the problem: The overall precision of the QED calculation is restricted, for typical LEP/SLC luminosity angular range, to 0.25%.
Typical running time: Efficiency for multiphoton sub-generator is 276 constant weight events and 756 variable weight events per one IBM 3090 CPU second.

1 Introduction

At high energy electron-positron scattering experiments like PETRA/PEP and presently LEP/SLC in order to translate the number of events into a cross-section one has to know the luminosity of the accelerator at the interaction point. The luminosity is deduced from measuring the number of events for one (or more) scattering process(es) for which the integrated cross-section is safely calculable from the theory. The e^+e^- elastic scattering, so-called Bhabha process, is a very good candidate because in the range of a few degrees it is dominated by perfectly known physics, the exchange of a t-channel photon. It is therefore used in all $e^{\pm}$ colliders for luminosity measurement. Note that, in order to keep statistical errors small, the process used for luminosity measurement has to have a large enough cross-section. This condition is fulfilled in the Bhabha small-angle scattering process rather easily, by putting the detector at small enough angles.

The small-angle Bhabha process, as any high energy process with light charged fermions, is subject to relatively large, a few percent, QED radiative corrections. In the experimental luminosity measurement these corrections have to be calculated and their effect taken into account, in other words, removed from the luminosity experimental value.

1.1 Systematics of QED prediction

It should be strongly stressed that, if the above procedure of removing the QED effect is applied to any experimental quantity, then all uncertainties of the QED calculation enter immediately its systematic error. In the case of the recent LEP experiments the QED component in the luminosity systematic experimental error was as large as all the other apparatus-related systematics.

If the QED uncertainty in luminosity measurement is so important the immediate questions are: where does it come from and what are the methods of reducing it? The easiest way to answer the first question is to enumerate the typical components in the total QED uncertainty in any calculation of the QED radiative corrections. We divide them into two equally important groups. The first group forms the QED uncertainty which we call collectively *technical precision* and by this we understand all kinds of errors due to numerical approximations and mistakes in the analytical/numerical calculations, including rounding errors, programming bugs, random number effects, etc., but not the higher order effects, new physics, etc. This component is inherently present in the calculation because even analytical calculations are finally encoded in the computer programs. In fact in the Bhabha scattering the numerical phase space integration of the Monte Carlo type is unavoidable owing to experimental cuts and acceptance. The second component of QED uncertainty we call *physical precision* and by this we understand higher order effects due to truncation of the perturbative series, effects due to neglecting some Feynman diagrams or sub-leading-logarithmic terms in the QED calculation. The entire QED uncertainty is the sum of the technical and physical precision (in quadrature). We refer the reader, for more detailed discussion on systematic errors of QED calculations, to Ref. [1] in the general case and to Refs. [2, 3] in the small-angle Bhabha case.

What are the means of calculating and eventually reducing the QED uncertainty? The essential prescription for estimation and reduction of the *technical* precision is the following: try to calculate the cross-sections and/or differential distributions twice or three times using calculational techniques as different as possible – take the difference between results as an estimate of the technical precision and work hard to reduce it. The other simple methods like varying the input parameter in the calculation, changing random number generators, etc. should be used routinely. Good examples of calculating technical precision are in Ref. [4], where the phase space integration for small-angle Bhabha's was done (for simplified cuts) once with the Monte Carlo method and a second time semi-analytically (combination of analytical and Gauss integrations), and in Ref. [3] where the integration was done using two independent Monte Carlo programs. In both cases numerical results from the two methods differed by only 0.02% and this was taken as the technical precision of the calculation.

The only acceptable source of the *physical* precision is the calculation of higher order effects and neglected terms (especially if we suspect them to be sizeable) or some kind of estimation of the upper limit on them (this may be reasonable in the case of contributions which are expected to be very small). This may often require developing new calculation methods and new Monte Carlo programs as well. Needless to say the solid estimate of the technical and physical precision of the QED radiative correction, as specified above, may be more work than calculation of the correction itself! This is, however, unavoidable because in the case where QED uncertainty is taken as a component of the experimental systematic error (removing QED from data) any kind of not-well-founded guesswork on the QED uncertainty cannot be accepted. It would make the experimental data (with removed QED) worthless.

How well do we know QED corrections and their uncertainty in the small-angle Bhabha ($\vartheta < 10^\circ$) scattering? At PETRA/PEP $e^\pm$ colliders the Monte Carlo program OLDBAB of Ref. [5] and the related works [6, 7] were used to calculate the $\mathcal{O}(\alpha)$ QED correction. They were typically of the order of a few percent (excluding vacuum polarization) and it was not clear what their uncertainty was, neither physical nor technical, most probably it was about[1] 1%. Such a precision, at the time, was perhaps enough as compared with the experimental precision of 2% or more. With the advent of SLC and LEP, the experimental precision in small-angle Bhabha and luminosity measurement improved quickly down to 1% and the best

[1] Note that the later papers [8] did not bring anything new to the question of precision of small-angle Bhabha calculation. They mainly concentrate on adding Z-exchange in large angles.

present measurements quote 0.6% experimental error.

1.2 Reduction of QED systematics in luminosity

In view of the fact that the QED uncertainty of 1% in the luminosity measurement at LEP/SLC became unacceptably high the authors of this paper have undertaken the task of reducing it to 0.25%. This was achieved in the series of three recent papers [4, 3, 2] and the Monte Carlo program presented in this paper is the product of these works and of one earlier work of Ref. [9]. Let us briefly summarize these works before we explain how the BHLUMI 2.01 Monte Carlo program represents them. We have started our work with the *necessary* step of reducing technical precision of the $\mathcal{O}(\alpha)$ QED calculations which were already in use in PETRA/PEP. In Ref. [4] we have done a very careful comparison of the Monte Carlo generator OLDBAB [5] with the completely new semi-analytical calculation and we have established the technical precision of this QED $\mathcal{O}(\alpha)$ calculation to be at the unprecedented level of 0.02%! In order to get this technical precision we had to modify the OLDBAB program. The modified version of OLDBAB is available in the present program under the modified name OLDBIS. Let us stress that although paper [4] did not directly bring anything new on higher order QED calculations, it has nevertheless brought *essential progress* in solving the problem of reducing uncertainty of the QED corrections in luminosity measurement. We are able to say whether a given Monte Carlo or other calculation *really* represents the QED $\mathcal{O}(\alpha)$ answer[2] within technical precision of 0.02%! In other words Ref. [4] provides a baseline (reference) calculation for any precise QED calculation in small-angle Bhabha scattering.

Once we knew what the precise $\mathcal{O}(\alpha)$ QED answer was, we could ask about the higher order corrections. To do the entire $\mathcal{O}(\alpha^2)$ would be a long calculation and in fact not a very economical way of improving on precision. One knows in advance that only the dominant part proportional to $\alpha^2 \ln^2(|t_0|/m_e^2)$ is really significant.[3] This part is relatively easy to calculate. (Note that $L = \ln(|t_0|/m_e^2)$ is the so-called big logarithm involving the typical t-channel transfer t_0.) In Ref. [3] the $\mathcal{O}(\alpha^2 L^2)$ leading-logarithmic correction to small-angle Bhabha's for semi-realistic cuts was calculated. The same very good technical precision of 0.02% was achieved. In this case, the reliable estimate of the technical precision was obtained by comparing two Monte Carlo leading-logarithmic (LL) calculations. The additional cross-check with a semi-analytical calculation for simplified (unrealistic) structure functions was also done and confirmed the above technical precision. We have also provided the LEP/SLC collaborations with the Monte Carlo event generator LUMLOG which allows one to calculate the $\mathcal{O}(\alpha^2 L^2)$ correction for arbitrary cuts/acceptance. This program is included in the present version of BHLUMI. The leading-logarithmic pro-

[2]This check was done for BABAMC program [8] by all LEP experimental collaborations, see Ref. [3].

[3]As was pointed out in Ref. [10] the leading-logarithmic third-order contributions are equally important as the second-order sub-leading contributions.

gram LUMLOG is based on structure functions and its QED perturbative content in fact extends to the third-order $\mathcal{O}(\alpha^3 L^3)$! It has Yennie-Frautschi-Suura (YFS) exponentiation of the latter, optionally. Its kinematics is based on strict collinearity of the bremsstrahlung photon emission. Furthermore, since in the typical luminosity measurement the final-state electron is measured calorimetrically (no distinction between electrons and photons in the detector) the LL final-state correction is exactly zero due to the Lee-Kinoshita-Nauenberg theorem. This is the reason why it was not necessary to include it in LUMLOG. In Ref. [3] we have also given a recipe for how one combines the $\mathcal{O}(\alpha)$ and $\mathcal{O}(\alpha^2 L^2)$ Monte Carlo results in order to reach the overall QED precision of 0.3%. We shall refer to this recipe in the following as the OLDBIS+LUMLOG solution. The main component in the above QED uncertainty was 0.2% due to $\mathcal{O}(\alpha^2 L)$ corrections and 0.1% due to light pair production. The t-channel vacuum polarization (0.08%) uncertainty was not included in this QED error.

The OLDBIS+LUMLOG recipe is a perfectly reasonable QED perturbative prescription but has two disadvantages: it is not represented by the single Monte Carlo event generator and it does not include exponentiation, i.e. summation of the soft photon effects to infinite order. The exponentiation technique is known to reduce significantly higher order effects [11, 10]. In fact OLDBIS and LUMLOG were from the beginning meant as tests of the BHLUMI program [9] which is a full-scale multi-photon event generator with the $\mathcal{O}(\alpha)$ Yennie-Frautschi-Suura exponentiation [12]. The 1% precision estimate of the early version of BHLUMI was obtained by comparing its results with non-exponentiated $\mathcal{O}(\alpha)$ calculations [9]. It was clear from the beginning that its actual precision could be better but there were no additional programs of comparable quality to prove this statement. In paper [2] the systematic comparisons of BHLUMI with the OLDBIS+LUMLOG combination gave a 0.25% overall precision estimate for the present version of BHLUMI.[4] This estimate includes the vacuum polarization uncertainty. Note that at present the 0.25% uncertainty of BHLUMI from paper [2] includes technical precision. The first results of an independent calculation of technical precision of BHLUMI based on the exponentiated version of the semi-analytical calculation of Ref. [4] were presented in Ref. [13] (this work needs to be completed). Note that on the way from BHLUMI version 1.xx to the present version 2.01 a lot of technical improvements were done which probably improve its technical precision significantly, notably, the 0.5% bias due to misuse of random number generators was removed and numerical instabilities in the matrix element calculation due to smallness of electron mass and of the electron scattering angle were cured, see also next Section.

How can the present program help in calculating QED corrections and their uncertainty in small-angle Bhabha luminosity measurement?

[4]This may look as a kind of paradox that $\mathcal{O}(\alpha)_{exp}$ BHLUMI is as precise as the $\mathcal{O}(\alpha^2)$ OLDBIS+LUMLOG solution but the key point [2] is that due to exponentiation BHLUMI includes almost all of the $\mathcal{O}(\alpha^2 L^2)$ correction [9] and (most probably) of $\mathcal{O}(\alpha^2 L)$ as well.

- The OLDBIS sub-generator may be used to check if any $\mathcal{O}(\alpha)$ Monte Carlo or analytical calculation really represents the QED $\mathcal{O}(\alpha)$ calculation in small-angle Bhabha's to within 0.02% technical precision. It embodies the high precision numerical benchmark of Ref. [4] for $\mathcal{O}(\alpha)$ QED. It may also serve to calculate the contribution from up-down interference which is very small and is often neglected; nevertheless, it should be kept under control because it enters into QED systematics.

- The LUMLOG sub-generator together with OLDBIS or any other $\mathcal{O}(\alpha)$ Monte Carlo generator may be used for implementing the OLDBIS+LUMLOG recipe of Ref. [3]. It is also a source of estimate of the light pair contribution. The LUMLOG+OLDBIS solution is the test of the BHLUM2 multiphoton sub-generator and we recommend comparing them again, following Ref. [2], for a given experimental set-up, especially if the experimental error is 0.5% or less.

- The BHLUM2 multi-photon sub-generator is the basic Monte Carlo generator of the presented program and it should be used for removing QED corrections from luminosity measurement. It is well suited for detector simulation. It features Yennie-Frautschi-Suura exponentiation – the efficient technique of reducing higher order corrections.

1.3 YFS exponentiation and Monte Carlo

In the following we shall describe briefly the Yennie-Frautschi-Suura (YFS) exponentiation and explain the role of the Monte Carlo technique in the practical calculations in the YFS framework.

The Yennie-Frautschi-Suura (YFS) exponentiation was formulated in the classic paper [12] and it represents *exclusive exponentiation*, not to be confused with the often practised *inclusive exponentiation*. We refer the reader to Ref. [14] for a more detailed explanation of this and many other aspects of exponentiation. In brief, exponentiation is the method of summing up to infinite order the contributions from soft virtual and real photons. In Ref. [12] all soft photon contributions/divergences were carefully analysed in the standard Feynman-diagrams based framework of QED, taking care of renormalizability (ultraviolet divergences) and gauge invariance. It is *exclusive* because isolation and summation (to infinite order) of the infrared divergent parts is done on exclusive differential cross-sections (in fact on scattering amplitudes) with an arbitrary number of *real* photons. The inclusive exponentiation was developed in order to avoid the difficult phase space integration in full YFS exponentiation. It amounts to *ad hoc* modifications of the one-dimensional inclusive distributions taking as a guide some partial results from the full YFS exponentiation, see examples in Ref. [14].[5].

[5]The inclusive *ad-hoc* exponentiation is in most cases done in $\mathcal{O}(\alpha)$, see [15], later it was done in $\mathcal{O}(\alpha^2)$ [16], and recently the $\mathcal{O}(\alpha^3)$ examples were constructed [17].

Monte Carlo implementation of the YFS exclusive exponentiation was started in Refs. [18] and [19] and the first Monte Carlo program for second-order initial state radiation YFS exponentiation was published in Ref. [20] which is currently included in the Monte Carlo program KORALZ [21]. The first Monte Carlo implementation of the YFS exponentiation for the t-channel process was described in Ref. [9]. There exists also an unpublished version of the YFS-type Monte Carlo program YFS3 for initial and final state bremsstrahlung (second order) for Z production at LEP/SLC [22], see also the recent effort in Ref. [23] in this direction. Why use the Monte Carlo method for YFS exponentiation? The reason is twofold: First of all, the original YFS inclusive exponentiation involves integrations over multiphoton phase space – the numerical Monte Carlo integration method is practically the only one which is able to evaluate these multidimensional integrals[6], and secondly, with help of the Monte Carlo simulation one may take even the most complicated experimental cuts and acceptance into account in the data analysis. Note that here and in the following we have in mind *exact integration* (within statistical errors) over the multiphoton phase space.[7] In fact the advent of fast computers and the development of the Monte Carlo methods created the first chance of profiting fully from the YFS exponentiation scheme.

What are the profits from the YFS exponentiation? In the YFS exponentiation infrared contributions are correctly summed up to infinite order. The remaining non-divergent parts are calculated perturbatively order by order. The profit, apart from having, once and for all the correct soft photon limit, is that the order by order convergence of the integrated cross-sections and differential distributions is usually better than in the non-exponentiated version of the calculation; see Ref. [17] for discussion in the case of s-channel $e^{\pm}$ annihilation into the Z-resonance. For a t-channel dominated process this phenomenon of boosting perturbative convergence was shown in Ref. [2].

The above arguments show that it is worth while to exponentiate using the YFS method, let us therefore consider the question of how to do it and how to create a Monte Carlo integration/simulation program for the t-channel dominated process in which we are interested for luminosity purposes (the other immediate application will be ep scattering of the HERA type). Since the Monte Carlo method is slowly convergent, it is therefore essential to understand fully the structure of the peaks in the integrand and to remove them (effectively) by the appropriate change of the variables. This means that one has to invent such variables in which these singularities are simple enough (like $\sim 1/x$) to be integrate by hand. In the small-angle case the basic singularities are the very strong $1/t^2$ photon exchange peak and

[6]Only the simplest case when all photons are very soft can be treated analytically. Some integrations can be done analytically in the leading-logarithmic approximation [12].

[7]It is often the case in the QCD calculations that part of the definition of differential/total cross-section is hard-wired into the Monte Carlo phase space integration algorithm such that the phase space and the matrix element are impossible to separate. In our case we follow the standard method "phase space times matrix element squared" as in the usual order by order QED calculations.

a bit milder bremsstrahlung peaks $1/(kp_1)(kp_2)$ for all real photons. To complicate things even more, the conservation of the total four momentum has to be taken into account. The appropriate variables were invented during the construction of the Monte Carlo program for simulation of the QED bremsstrahlung from the electron line in the *ep* scattering (HERA) [24] and their generalization to multiphoton case was presented in Ref. [9]. In fact, they are variant of the Sudakov light-cone variables known for a long time [25]. In the new variables the phase space together with the infrared singularities due to real photons appeared to be treatable by the standard Monte Carlo methods.

The rest of the paper is organized as follows. In Section 2 we shall describe essential ingredients in the Monte Carlo algorithm and QED matrix element in all three sub-generators. In Section 3 we list most of the subprograms and describe briefly their role. In Section 4 we provide instructions on how to use the program and provide detailed description of the input/output. Short conclusions terminate the paper.

2 Monte Carlo algorithm

In this Section we shall describe in detail the multiphoton Monte Carlo event sub-generator BHLUM2 of BHLUMI 2.01, which features $\mathcal{O}(\alpha)$ Yennie-Frautschi-Suura exponentiation. We shall give a complete technical description of its Monte Carlo algorithm and explain differences from the previous versions BHLUMI 1.xx. We shall also give a brief description of the Monte Carlo algorithm of LUMLOG and a short comment on the small but important modification in the algorithm of OLDBIS with respect to OLDBAB.

The Monte Carlo algorithm of BHLUM2, multiphoton generator described first in Ref. [9], represents the *minimal solution* of the problem of the phase space integration over bremsstrahlung singularities in the presence of the strong t-channel singularity $1/t^2$. It is minimal because it is based on the choice of the phase space variables in which the "bremsstrahlung phase space" = (squared matrix element modulus)$\times$(phase space) is completely flat (constant) and in the Monte Carlo algorithm it is not possible to do better! It was possible to achieve this by using Sudakov-type variables. Sudakov has discovered that in these variables the integration over "bremsstrahlung phase space" amounts simply to calculating (in leading-logarithmic approximation) the area of the integration domain which is typically a combination of rectangles and/or triangles. With respect to Sudakov we improved the method a lot: in transforming the "bremsstrahlung phase space" to his variables Sudakov neglects $1/\ln(|t|/m^2)$ terms (LL approximation) while in our variant we treat this distribution *exactly*, i.e. dropping only $m_e^2/|t|$ terms.[8] Achieving complete flatness may look in the actual calculations a bit like a miracle but the essential

[8]The above mentioned rectangles and triangles in Sudakov variables become a bit oval in our exact calculations. In the Monte Carlo this is taken into account.

reason for this to be true is that for (almost) massless fermions the bremsstrahlung "integration element"

$$d\tilde{\omega} = \frac{d^3k}{k^0}\,\frac{\alpha}{4\pi^2}\,\frac{2p_1p_2}{(p_1k)(p_2k)} \tag{1}$$

is not only Lorentz invariant, but also scale-invariant. Our method of integration over t-channel "bremsstrahlung phase space" apart from being minimal and natural is also *universal.* It is universal because it can be applied to any kind of QED problem. For example, it is not necessarily bound to exponentiation, i.e. non-exponentiated[9] $\mathcal{O}(\alpha)$ (see. Ref. [24, 9]) or $\mathcal{O}(\alpha^2)$ Monte Carlo calculations can be done with our method. Furthermore, it extends also to multiphoton bremsstrahlung in the final-state fermion-pair system [22]. Let us finally stress that although the Monte Carlo algorithm presented here is minimal for the particular problem, nevertheless, it is currently (in our opinion) the most sophisticated Monte Carlo algorithm for the QED bremsstrahlung calculation.

2.1 BHLUM2 MASTER FORMULA

The complete *master formula* for the $\mathcal{O}(\alpha^1)$ exponentiated total cross-section for the process $e^+(p_1)+e^-(q_1)\rightarrow e^+(p_2)+e^-(q_2)+n\gamma(k_i)+n'\gamma(k'_{i'})$ as actually implemented in the BHLUMI 2.01 Monte Carlo program is the same as in Ref. [2] and it reads as follows[10]

$$\begin{aligned}
\sigma = &\sum_{n=0}^{\infty}\sum_{n'=0}^{\infty}\frac{1}{n!}\frac{1}{n'!}\int\frac{d^3q_2}{q_2^0}\,\frac{d^3p_2}{p_2^0}\,\delta^{(4)}\Big(p_1+q_1-p_2-q_2-\sum_{i=1}^{n}k_i-\sum_{i'=1}^{n'}k'_{i'}\Big)\\
&\exp\Big(Y(\Omega_1,p_1,p_2)+Y(\Omega_2,q_1,q_2)\Big)\\
&\int\prod_{i=1}^{n}\frac{d^3k_i}{k_i^0}\tilde{S}(p_1,p_2,k_i)(1-\Theta(\Omega_1;k_i))\int\prod_{j=1}^{n'}\frac{d^3k'_j}{k'^0_j}\tilde{S}(q_1,q_2,k'_j)(1-\Theta(\Omega_2;k'_j))\\
&\Big(\bar{\beta}_0^{(1)}(Q,p_1,p_2,q_1,q_2)+\sum_{i=1}^{n}\bar{\beta}_1^{(1)}(Q,p_1,p_2,q_1,q_2,k_i)/\tilde{S}(p_1,p_2,k_i)\\
&\qquad+\sum_{j=1}^{n'}\bar{\beta}_{1'}^{(1)}(Q,p_1,p_2,q_1,q_2,k'_j)/\tilde{S}(q_1,q_2,k'_j)\Big)\;\Xi_{\rm M.C.}(p_i,q_i,k_l,k'_m),
\end{aligned}\tag{2}$$

where $\tilde{S}(p_1,p_2,k) = -(\alpha/4\pi^2)((p_1/kp_1)-(p_2/kp_2))^2$ is the real photon infrared factor and

$$Y(\Omega,p_1,p_2) \;=\; 2\alpha\tilde{B}(\Omega,p_1,p_2)+2\alpha\Re B(p_1,p_2)$$

[9]The BHLUMI 1.xx included a non-exponentiated $\mathcal{O}(\alpha)$ sub-generator which was used for tests and it is not included here.

[10]Note that taking only $n+n'=0,1$ and expanding the form-factor $\exp(Y(\Omega_1)+Y(\Omega_2))$ one recovers the ordinary non-exponentiated $\mathcal{O}(\alpha^1)$ expression for the differential cross-sections. For instance, defining $\Omega_{1,2}$ by $k^0<\varepsilon\sqrt{s}/2$ in the laboratory frame one recovers exactly the Eq. (1) of Ref. [4].

$$= \quad -2\alpha \frac{1}{8\pi^2}\int \frac{d^3k}{k^0}\Theta(\Omega;k)\Big(\frac{p_1}{kp_1}-\frac{p_2}{kp_2}\Big)^2$$
$$+2\alpha\Re\int \frac{d^4k}{k^2}\frac{i}{(2\pi)^3}\Big(\frac{2p_1-k}{2kp_1-k^2}-\frac{2p_2-k}{2kp_2-k^2}\Big)^2 \qquad (3)$$

is the standard Yennie-Frautschi-Suura form factor [12]. It is infrared finite and $\Theta(\Omega;k)=1$ for $k\in\Omega$, and $\Theta(\Omega;k)=0$ for $k\notin\Omega$. The infrared Ω region includes the $k=0$ infrared point and its definition may implicitly involve the dependence on fermion four-momenta p_i and q_i [12]. None of the physically sensible results depends on the choice of Ω! The Ω domain is typically defined through the $k^0 < E_{\rm min}$ condition in a certain reference frame. (In fact, the program features two types of Ω but only one of them is in use, see later this Section.) We shall define $\Omega_{1,2}$ and give the corresponding explicit formula for the form-factors later, while describing the Monte Carlo algorithm.

The perturbative $\mathcal{O}(\alpha)$ QED matrix element is located in the $\bar\beta$'s which are[11]

$$\bar\beta_0^{(1)}(Q,p_1,p_2,q_1,q_2)=\bar\beta_0^{(0)}(Q,p_1,p_2,q_1,q_2)(1+2\delta_0+\delta_\gamma+\delta_Z), \qquad (4)$$
$$\delta_0 = 2\Re F_1(Q^2)-2\Re B(Q^2)=\frac{1}{2}\beta_t,\quad \beta_t=2\frac{\alpha}{\pi}\left(\ln\frac{|Q^2|}{m_e^2}-1\right), \qquad (5)$$
$$\bar\beta_0^{(0)}(Q,p_1,p_2,q_1,q_2)=\frac{2\alpha_r(t)^2}{s}\,\frac{(s^2+u^2+s_1^2+u_1^2)}{4t_pt_q}, \qquad (6)$$
$$\bar\beta_1^{(1)}(Q,p_1,p_2,q_1,q_2,k_i)=\frac{\alpha_r(t)^2}{2s}\frac{\alpha}{4\pi^2}\,D_1^{(1)}(Q,p_1,p_2,q_1,q_2,k_i)$$
$$-\tilde S(p_1,p_2,k)\,\bar\beta_0^{(0)}(Q,p_1,p_2,q_1,q_2), \qquad (7)$$
$$\bar\beta_{1'}^{(1)}(Q,p_1,p_2,q_1,q_2,k'_j)=\frac{\alpha_r(t)^2}{2s}\frac{\alpha}{4\pi^2}\,D_{1'}^{(1)}(Q,p_1,p_2,q_1,q_2,k'_j)$$
$$-\tilde S(q_1,q_2,k'_j)\,\bar\beta_0^{(0)}(Q,p_1,p_2,q_1,q_2), \qquad (8)$$
$$D_1^{(1)}(Q,p_1,p_2,q_1,q_2,k)=$$
$$\frac{1}{(kp_1)(kp_2)}\left\{\frac{s^2+u_1^2}{|t_q|}\left(1-\frac{2m_e^2}{|t_q|}\,\frac{kp_1}{kp_2}\right)+\frac{s_1^2+u^2}{|t_q|}\left(1-\frac{2m_e^2}{|t_q|}\,\frac{kp_2}{kp_1}\right)\right\}, \qquad (9)$$
$$D_{1'}^{(1)}(Q,p_1,p_2,q_1,q_2,k)=$$
$$\frac{1}{(kq_1)(kq_2)}\left\{\frac{s^2+u_1^2}{|t_p|}\left(1-\frac{2m_e^2}{|t_p|}\,\frac{kq_1}{kq_2}\right)+\frac{s_1^2+u^2}{|t_p|}\left(1-\frac{2m_e^2}{|t_p|}\,\frac{kq_2}{kq_1}\right)\right\}, \qquad (10)$$
$$t=Q^2=(p_2+\sum_{i=1}^{n}k_i-p_1)^2,\ t_p=-2p_1p_2,\ t_q=-2q_1q_2,$$
$$s=2p_1q_1,\ s_1=2p_2q_2,\ u=-2p_1q_2,\ u_1=-2q_1p_2.$$

[11]Note that in the analogous formula in Ref. [2] the expression for $\tilde\beta_1^{(1)}$ was distorted and factor 2 in front of δ_0 was omitted. The formula in the program was always correct so this does not have any consequences for numerical results in this paper.

We implement vacuum polarization through the QED running coupling constant $\alpha_r(t) = \alpha/|1 + \Pi(t)|$ at the proper $Q^2 = t$ scale. This takes into account the vacuum polarization correction in the $\mathcal{O}(\alpha^2 L^2)$, as was pointed out in Ref. [26]. The correction $\delta_\gamma = t/s$ is due to s-channel γ exchange and the correction δ_Z represents here the interference of the t-channel photon with the s-channel Z

$$\delta_Z = \frac{\alpha_r(s)}{\alpha_r(t)} \left(\frac{t}{s}\right) \frac{2s^2}{s^2 + (t+s)^2} \left(1 + \frac{t}{s}\right)^3 (v^2 + a^2)\, \Re\left(\frac{s}{s - M^2 + is\Gamma/M}\right), \qquad (11)$$

where $a = -1/(4 \sin\theta_W \cos\theta_W)$, $v = a(1 - 4\sin^2\theta_W)$, M and Γ are the usual coupling constants, mass and width of Z. We use $\sin^2\theta_W = 0.2306, M = 91.161$ GeV and $\Gamma = 2.534$ GeV and these values are already precise enough for the purpose of luminosity measurement. In the above two corrections we keep terms which are necessary for the precision $< 10^{-4}$ for angles $\vartheta < 10°$.

The main difference in the above QED matrix element with respect to BHLUMI 1.xx is the neglect of up-down interference. This contribution was found in Ref. [4] to be very small in small-angle Bhabha's, for $\vartheta < 6°$ it is generally below 0.02%. In any case, for the purpose of the discussion of the physical error the OLDBIS sub-generator will provide the value of this contribution for any cut or acceptance. Dropping up-down interference allows us to consider bremsstrahlung from upper e^+ and lower e^- fermion lines independently, and to simplify the multiphoton bremsstrahlung matrix element considerably. In the process of writing the $\mathcal{O}(\alpha)$ multiphoton matrix element in the YFS exponentiation it is necessary to extend (extrapolate) the single bremsstrahlung matrix element beyond the three-body phase space.[12] Instead of doing it by means of manipulating four-momenta arguments in the corresponding expressions, as in Refs. [20, 9] (so-called reduction procedure), we rather extrapolate the single bremsstrahlung matrix element expressed in terms of Mandelstam variables, see Eq. (2). This method gives almost the same numerical result while it leads to more compact and explicit expressions which are faster and numerically more stable in the computer evaluation. It should be stressed, however, that reinstalling up-down interference in the present program is possible and it would be rather straightforward – the basic Monte Carlo algorithm is already prepared for this, see later in this section. We did not do it because in the small-angle Bhabha we regard up-down interference as an unnecessary complication!

The function

$$\Xi_{\rm M.C.}(p_i, q_i, k_j) = \theta(|t| - |t_{\min}|)\theta(|t_{\max}| - |t|) \qquad (12)$$

defines phase space for events generated in the Monte Carlo run. The user's own experimental trigger $\Xi_{\rm exper.}$ is imposed later by the usual rejection method, see Section 4.4 for discussion of the practical choice of $t_{\rm min,max}$. Cross-sections and distributions obtained with $\Xi_{\rm exper.}$ do not and should not depend on the particular values of

[12]This extrapolation is inherent in any kind of exponentiation and owing to the fact that infrared singularities were subtracted and summed up to infinite order, see Refs. [12, 14] for more comments.

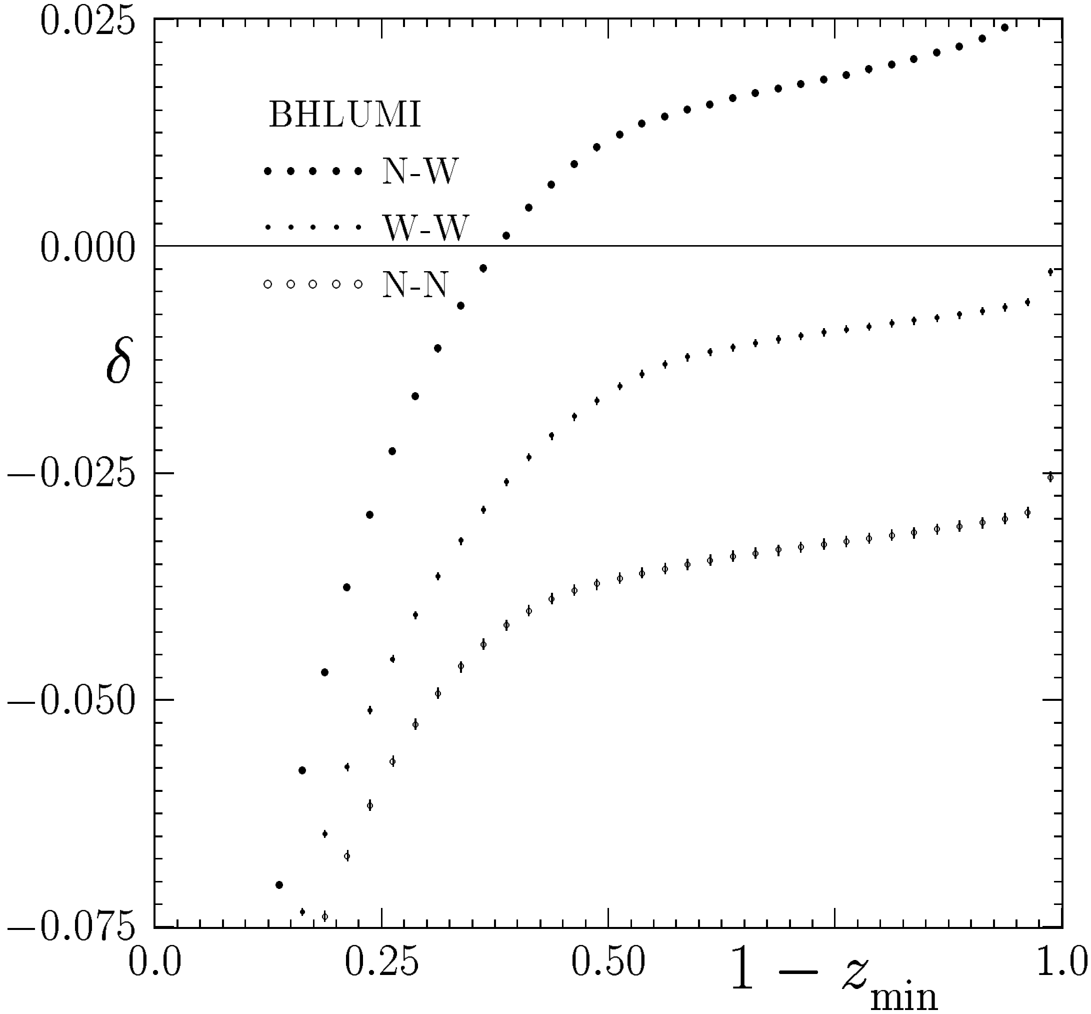

Figure 1: *The dependence of the total QED correction $\delta = \sigma/\sigma_{\rm Born} - 1$ on the energy cut $z_{\rm min}$ where σ is defined in eq. (2) and calculated with multiphoton sub-generator of the present program. (The vacuum polarization, Z and γ s-channel contribution are included.) Statistical Monte Carlo errors are indicated (they are usually below the size of dots). Monte Carlo sample includes $2.5 \cdot 10^7$ weighted events. Triggers N-N, W-W, N-W and the energy cut $z_{\rm min}$ are defined as in Ref. [2].*

$t_{\rm min,max}$. Note also that transfer t has physical meaning only if up-down interference is neglected and/or in the LL approximation. Otherwise it is an intermediate parameter in the Monte Carlo generation, being a complicated function of the photon momenta dependent on details of the Monte Carlo generation algorithm.

The total QED correction (with respect to Born which includes t-channel γ exchange only) corresponding to QED matrix element of eq. (2) is shown in Fig. 1. It is calculated with the present Monte Carlo program[13]. The correction δ is shown not only for the most interesting asymmetric acceptance function Ξ_{NW} but also for symmetric triggers Ξ_{NN} and Ξ_{WW}. These semi-realistic triggers and energy cut parameter $z_{\rm min}$ are defined in Ref. [2].

2.2 BHLUM2 Monte Carlo algorithm

The construction of the Monte Carlo algorithm involves two essential steps: the choice of proper integration variables and the simplifications of the integrand which are later compensated by appropriate rejection. To each simplification corresponds a well-defined component in the rejection weight.

Let us rewrite first Eq. (2) in a more compact notation

$$\sigma = \sum_{n=0}^{\infty}\sum_{n'=0}^{\infty} \frac{1}{n!}\frac{1}{n'!} \int \frac{d^3 q_2}{q_2^0}\, \frac{d^3 p_2}{p_2^0}\, \delta^{(4)}\Big(p_1+q_1-p_2-q_2-\sum_{i=1}^{n} k_i - \sum_{i'=1}^{n'} k_{i'}\Big)$$
$$\int \prod_{i=1}^{n} d\omega_i \int \prod_{i'=1}^{n'} d\omega'_{i'}\; \bar{\beta}_{0+1}\; \Xi_{\rm M.C.}\; e^{Y(\Omega_1)+Y(\Omega_2)}, \tag{13}$$

where

$$d\omega_i = \frac{d^3 k_i}{k_i^0} \tilde{S}(p_1,p_2,k_i)(1-\Theta(\Omega_1;k_i)) \tag{14}$$

and the rest of notation is self-explanatory.

The following crucial identity expressing introduction of the Sudakov-type variables in the phase space holds up to $m_e^2/|t|$ terms

$$\int \frac{d^3 q_2}{q_2^0}\, \frac{d^3 p_2}{p_2^0}\, \delta^{(4)}\Big(p_1+q_1-p_2-q_2-\sum_{i=1}^{n} k_i - \sum_{i'=1}^{n'} k_{i'}\Big) \int \prod_{i=1}^{n} d\omega_i \int \prod_{i'=1}^{n'} d\omega'_{i'}$$
$$\equiv \int_{|t_{\rm min}|}^{|t_{\rm max}|} \frac{d|t|}{s} \int_0^{2\pi} d\phi \int \prod_{i=1}^{n} d\omega_i \int \prod_{i'=1}^{n'} d\omega'_{i'}\; \left|\frac{t_p}{t}\right|\; \left|\frac{t_q}{t}\right|\; \theta(s_1)\; \Theta_\beta\; \Theta'_\beta, \tag{15}$$

where the bremsstrahlung integration elements ω_i on the left-hand side of the identity are expressed in terms of four-momenta, as in Eq. (14), while on the right-hand

[13]In Fig. 1 of Ref. [2] the same plot is not correct. The whole plot of δ is shifted there about -1% beacuse of bug in the plotting program (independent of the Monte Carlo program). The value of δ in Table 2a and other plots in Ref. [2] are not affected.

side they are functions of the new variables:

$$d\omega_i = \frac{\alpha}{\pi}\,\frac{\tilde{\alpha}_i\tilde{\beta}_i\,(1-\Theta(\Omega_1))\;d\tilde{\alpha}_i d\tilde{\beta}_i}{(\tilde{\alpha}_i+\delta_p\tilde{\beta}_i)^2(\tilde{\beta}_i+\delta_p\tilde{\alpha}_i)^2},\;\; \delta_p = \frac{m_e^2}{|t_p|},\;\; 0<\tilde{\alpha}_i,\tilde{\beta}_i<1,\;\; \Theta_\beta = \theta(1-\sum\tilde{\beta}_i), \tag{16}$$

and similarly

$$d\omega'_i = \frac{\alpha}{\pi}\,\frac{\tilde{\alpha}'_i\tilde{\beta}'_i\,(1-\Theta(\Omega_2))\;d\tilde{\alpha}'_i d\tilde{\beta}'_i}{(\tilde{\alpha}'_i+\delta_q\tilde{\beta}'_i)^2(\tilde{\beta}'_i+\delta_q\tilde{\alpha}'_i)^2},\;\; \delta_q = \frac{m_e^2}{|t_q|},\;\; 0<\tilde{\alpha}'_i,\tilde{\beta}'_i<1,\;\; \Theta'_\beta = \theta(1-\sum\tilde{\beta}'_i). \tag{17}$$

We define variables $\tilde{\alpha}_i$ $\tilde{\beta}_i$ exactly as in Eqs. (12-14) of Ref. [9], in terms of the momenta in the corresponding t-channel rest frame QRS_p where $p_1^0 = p_2^0 = E_p$ and $\vec{p}_1+\vec{p}_2 = 0$. For the convenience of the reader let us recall these definitions

$$\begin{aligned}
&k_i^0 = (\alpha_i+\beta_i)E_p, \quad k_i^3 = (-\alpha_i+\beta_i)E_p,\\
&k_i^1 = k_T\cos\phi_i, \quad k_i^2 = k_T\sin\phi_i, \quad k_T = 2E_p\sqrt{\alpha_i\beta_i},\\
&\alpha_i = \tilde{\alpha}_i K_p, \quad \beta_i = \tilde{\beta}_i K_p, \quad K_p = \Big(1-\sum_{j=1}^{n}\tilde{\beta}_j\Big)^{-1} = \frac{p_1(p_2+\sum_{j=1}^{n}k_j)}{p_1p_2},
\end{aligned} \tag{18}$$

see Ref. [9] for more details. Similarly, we define the variables $\tilde{\alpha}'_i$ $\tilde{\beta}'_i$ in the analogous way in the rest frame of the q_1+q_2 system, QRS_q. Note that, at first sight, Eqs. (16) and (17) differ from ω_i in Eq. (11) of Ref. [9]. They are, however, identical up to $m_e^2/|t|$ terms (in the integrated cross-section). The present form helps to fight numerical instabilities due to rounding errors at small photon angles. Let us stress that variables $t_p = t_p(t,\tilde{\alpha}_i,\tilde{\beta}_i)$, $t_q = t_q(t,\tilde{\alpha}'_i,\tilde{\beta}'_i)$, and $s_1 = s_1(s,t,\tilde{\alpha}_i,\tilde{\beta}_i,\tilde{\alpha}_i,\tilde{\beta}_i)$ are known but complicated functions of the bremsstrahlung variables. The $\theta(s_1)$ condition reflects the fact that some high values of $\tilde{\alpha}$'s and $\tilde{\beta}$'s, close to one, lead formally to negative s_1, i.e. outside the phase space.[14]

With the above transformation to new variables the master formula of Eq. (2) takes the new form

$$\sigma = \sum_{n=0}^{\infty}\sum_{n'=0}^{\infty}\frac{1}{n!}\frac{1}{n'!}\int_{|t_{\min}|}^{|t_{\max}|}\frac{d|t|}{s}\int_0^{2\pi}d\phi\left|\frac{t_p}{t}\right|\left|\frac{t_q}{t}\right|\int\prod_{i=1}^{n}d\omega_i\prod_{i'=1}^{n'}d\omega'_{i'}\;\theta(s_1)\;\bar{\beta}_{0+1}\;e^{Y(\Omega_1)+Y(\Omega_2)}, \tag{19}$$

which will be taken as a starting point for constructing the Monte Carlo integration-/simulation algorithm. Is the above new phase space parametrization rendering the integrand flat? Apart from the region very close to the edges of the phase space where there are some additional helicity zeros, the integrand behaves essentially like $\simeq (dt/t^2)\prod_i(d\tilde{\alpha}_i/\tilde{\alpha}_i)(d\tilde{\beta}_i/\tilde{\beta}_i) = d(1/t)\,\prod_i d(\ln\tilde{\alpha}_i)d(\ln\tilde{\beta}_i)$, i.e. the simple logarithmic

[14] In fact, in the program we use rather $\theta(s_1-4m_e^2)$.

substitution makes it constant and the t-integration clearly factorizes off! All additional variation hidden in $\bar{\beta}_{0+1}$ is rather mild and can be easily introduced in the Monte Carlo by weighting or rejection.

In the new phase space parametrization we are ready, now, to define the soft-photon regions $\Omega_{1,2}$ and calculate the corresponding YFS form factors $Y(\Omega_{1,2})$. We shall define them directly in terms of the Sudakov-type variables. Let us do it in detail for Ω_1, the case of Ω_2 is completely analogous. It is easy to find the expression for $Y(\Omega_1')$ in the case of the conventional spherical boundary in the QRS_p frame, see above, defined by $2k^0/\sqrt{|t_p|} < \delta$, or in terms of our intermediate variables, by $\alpha_i + \beta_i < \delta$. We simply take the s-channel formula, Eq. (17) of Ref. [19] (or Eq. (2.5) in Ref. [20]) and add $-\pi^2/2$ due to analytical continuation in the virtual $2\Re B$ function from s to t channel. The result is

$$Y(\Omega_1', p_1, p_2) = \frac{\alpha}{\pi}\left\{2\Big(\ln\frac{|t_p|}{m_e^2} - 1\Big)\ln\delta + \frac{1}{2}\ln\frac{|t_p|}{m_e^2} - 1 - \frac{\pi^2}{6}\right\}. \tag{20}$$

The rescaling transformation $(\alpha_i, \beta_i) \to (\tilde{\alpha}_i, \tilde{\beta}_i)$ translates this condition into $\tilde{\alpha}_i + \tilde{\beta}_i < \delta/K_p$. The actual generation of real photons is done more conveniently in the "rectangular" region $1 > \max(\tilde{\alpha}_i, \tilde{\beta}_i) > \Delta$, $\Delta = \delta/K_p$, and our soft-photon region Ω_1 in the form-factor calculation is defined as $\max(\tilde{\alpha}_i, \tilde{\beta}_i) < \Delta$. The real photon integral over the triangle area $\max(\tilde{\alpha}_i, \tilde{\beta}_i) < \Delta$, $(\tilde{\alpha}_i + \tilde{\beta}_i) > \Delta$ adds to Eq. (20) the contribution $+(\alpha/\pi)(\pi^2/6)$. Summarizing, the corresponding YFS formfactor for Ω_1 reads as follows[15]

$$Y(\Omega_1, p_1, p_2) = \frac{\alpha}{\pi}\left\{2\Big(\ln\frac{2p_1p_2}{m_e^2} - 1\Big)\ln\Delta_p + \frac{1}{2}\ln\frac{2p_1p_2}{m_e^2} - 1\right\}, \quad \Delta_p = K_p\Delta. \tag{21}$$

The definition of Ω_2 and the corresponding form factor $Y(\Omega_2)$ is completely analogous. It is clear that the two soft regions $\Omega_{1,2}$ overlap partly and they do not represent anything simple in the laboratory system. This solution is satisfactory and economical as long the photon emission is considered independently from e^+ and e^- fermion lines, i.e. as long as we neglect up-down interferences. In fact there exists an elegant method (better than rejection in Ref. [9]) of imposing common infrared boundary on soft photons in the laboratory system. It is already implemented in this program, see later in this Section. This may be helpful if, one day, the up-down interferences are re-introduced for the large-angle Bhabha scattering extension of the program.

The Monte Carlo algorithm is based on the rejection method. The true differential density

$$d\rho = \frac{1}{n!n'!}d\phi\frac{dt}{s}\left|\frac{t_p}{t}\right|\left|\frac{t_q}{t}\right| e^{Y(\Omega_1)+Y(\Omega_2)}\prod_{i=1}^{n} d\omega_i \prod_{i'=1}^{n'} d\omega'_{i'}\ \theta(s_1)\ \bar{\beta}_{0+1}, \tag{22}$$

[15] Note that the $K = K_pK_q$ dilatation factor was by mistake omitted in Ref. [2]. It was always present in the program, therefore, this omission is of no consequence for any numerical result in this paper.

is temporarily replaced by the approximate "crude" density $d\rho_0$, which is simple enough to be integrated by hand and to be generated with the simple Monte Carlo methods [27], and all effects of this simplification are readily removed by rejecting events according to weight

$$W = \frac{d\rho}{d\rho_0}. \tag{23}$$

The rejection is in fact not necessary and one may work with weighted events as well. In any case the total cross-section is given, up to statistical error, by

$$\sigma =< W > \sigma_0 =< W > \int d\rho_0, \tag{24}$$

where the average weight $< W >$ is calculated numerically and the crude cross-section σ_0 is known analytically. In practice the above substitution $\rho \to \rho_0$ is done in a few steps and each factor $W^{(i)}$ in the total weight

$$W = \prod_{k=1}^{5} W^{(k)} \tag{25}$$

corresponds to one simplification step. In the following we shall describe in detail all stages of simplification in our case, giving explicitly the corresponding weight factors.

1. We start the simplification with the integration elements $d\omega_i$ and $d\omega_i'$. Let us discuss the case of $d\omega_i$. First, we get rid of dependence on $t_p = t_p(t, \tilde{\alpha}_i, \tilde{\beta}_i)$ in δ_p by replacing it as follows

$$\delta_p \to \frac{m_e^2}{s} = \delta_s. \tag{26}$$

Let us remark in passing that for calculation at very small-angles $\theta_{\min} < 1^o$ it is possible and profitable to use $|t_{\max}|$ instead of s in the above replacement, see Section 4 for more details. At the same time we set

$$\frac{\tilde{\alpha}_i\tilde{\beta}_i}{(\tilde{\alpha}_i + \delta_p\tilde{\beta}_i)(\tilde{\beta}_i + \delta_p\tilde{\alpha}_i)} \to 1, \qquad \left|\frac{t_p}{t}\right| \to 1. \tag{27}$$

Summarizing, to all these replacements[16]

$$\left|\frac{t_p}{t}\right| \prod_{i=1,n}^{n} d\omega_i \longrightarrow \prod_{i=1,n}^{n} d\tilde{\omega}_i, \quad d\tilde{\omega}_i = \frac{\alpha}{\pi}\frac{d\tilde{\alpha}_i d\tilde{\beta}_i \theta(\max(\tilde{\alpha}_i, \tilde{\beta}_i) - \Delta)}{(\tilde{\alpha}_i + \delta_s\tilde{\beta}_i)(\tilde{\beta}_i + \delta_s\tilde{\alpha}_i)}, \tag{28}$$

corresponds the "mass weight" for the electron line

$$W^{(1)} = \left|\frac{t_p}{t}\right| \prod_{i=1,n}^{n} W_i^{(1)}, \qquad W_i^{(1)} = \frac{\tilde{\alpha}_i\tilde{\beta}_i(\tilde{\alpha}_i + \delta_s\tilde{\beta}_i)(\tilde{\beta}_i + \delta_s\tilde{\alpha}_i)}{(\tilde{\alpha}_i + \delta_p\tilde{\beta}_i)^2(\tilde{\beta}_i + \delta_p\tilde{\alpha}_i)^2}. \tag{29}$$

[16]Note that $\theta(\max(\tilde{\alpha}_i, \tilde{\beta}_i) - \Delta)$ enters also in $d\tilde{\omega}_i$ and therefore drops out from the weights.

2. The analogous modifications for the positron line lead to the second mass weight

$$W^{(2)} = \left|\frac{t_q}{t}\right| \prod_{i'=1,n'}^{n'} W_{i'}^{(2)}, \qquad W_{i'}^{(2)} = \frac{\tilde{\alpha}_{i'}\tilde{\beta}_{i'}(\tilde{\alpha}_{i'} + \delta_s\tilde{\beta}_{i'})(\tilde{\beta}_{i'} + \delta_s\tilde{\alpha}_{i'})}{(\tilde{\alpha}_{i'} + \delta_q\tilde{\beta}_{i'})^2(\tilde{\beta}_{i'} + \delta_q\tilde{\alpha}_{i'})^2}, \tag{30}$$

3. Next, in order to get rid of the complicated dependence $s_1(s, t, \tilde{\alpha}_i, \tilde{\beta}_i, \tilde{\alpha}_i, \tilde{\beta}_i)$ we set $\theta(s_1)\Theta_\beta\Theta'_\beta \to 1$ and the corresponding weight is simply

$$W^{(3)} = \theta(s_1)\Theta_\beta\Theta'_\beta. \tag{31}$$

4. For the same purpose we make the following simplifications in the YFS form factors

$$Y(\Omega_1) + Y(\Omega_2) \longrightarrow 2Y_\Delta = -4\frac{\alpha}{\pi}\ln\delta_s\ln\Delta = 4\frac{\alpha}{\pi}\ln\frac{s}{m_e^2}\ln\Delta. \tag{32}$$

   The corresponding "form factor weight" reads as follows

$$W^{(4)} = \exp(Y(\Omega_1) + Y(\Omega_2) - 2Y_\Delta). \tag{33}$$

   This replacement also fixes conveniently the normalization of the average total weight (we choose it such that $< W > \simeq 1$).

5. Finally, the last replacement of the series of $\tilde{\beta}$'s by a Born-like distribution

$$\bar{\beta}_{0+1} \longrightarrow b_0 = \frac{2\alpha^2 s}{t^2} \tag{34}$$

   leads to the important "model component weight"

$$W^{(5)} = \frac{1}{b_0}\bar{\beta}_{0+1}. \tag{35}$$

The crude Monte Carlo cross-section resulting from the above five simplifications is defined as follows

$$\sigma_0 = \int d\rho_0 = \sum_{n=0}^{\infty}\sum_{n'=0}^{\infty}\frac{1}{n!}\frac{1}{n'!}\int_{|t_{\min}|}^{|t_{\max}|}\frac{d|t|}{s}\int_0^{2\pi} d\phi e^{2Y_\Delta}\int\prod_{i=1}^{n} d\tilde{\omega}_i \prod_{i'=1}^{n'} d\tilde{\omega}'_{i'}\, b_0. \tag{36}$$

In view of $\int d\tilde{\omega}_i = 2\frac{\alpha}{\pi}\ln\delta_s\ln\Delta$ all integration and summation can be done analytically leading to the very simple result

$$\sigma_0 = 4\alpha^2\pi\left(\frac{1}{|t_{\min}|} - \frac{1}{|t_{\max}|}\right)e^{2Y_\Delta}\exp\left[4\frac{\alpha}{\pi}\ln\delta_s\ln\Delta\right] = 4\alpha^2\pi\left(\frac{1}{|t_{\min}|} - \frac{1}{|t_{\max}|}\right). \tag{37}$$

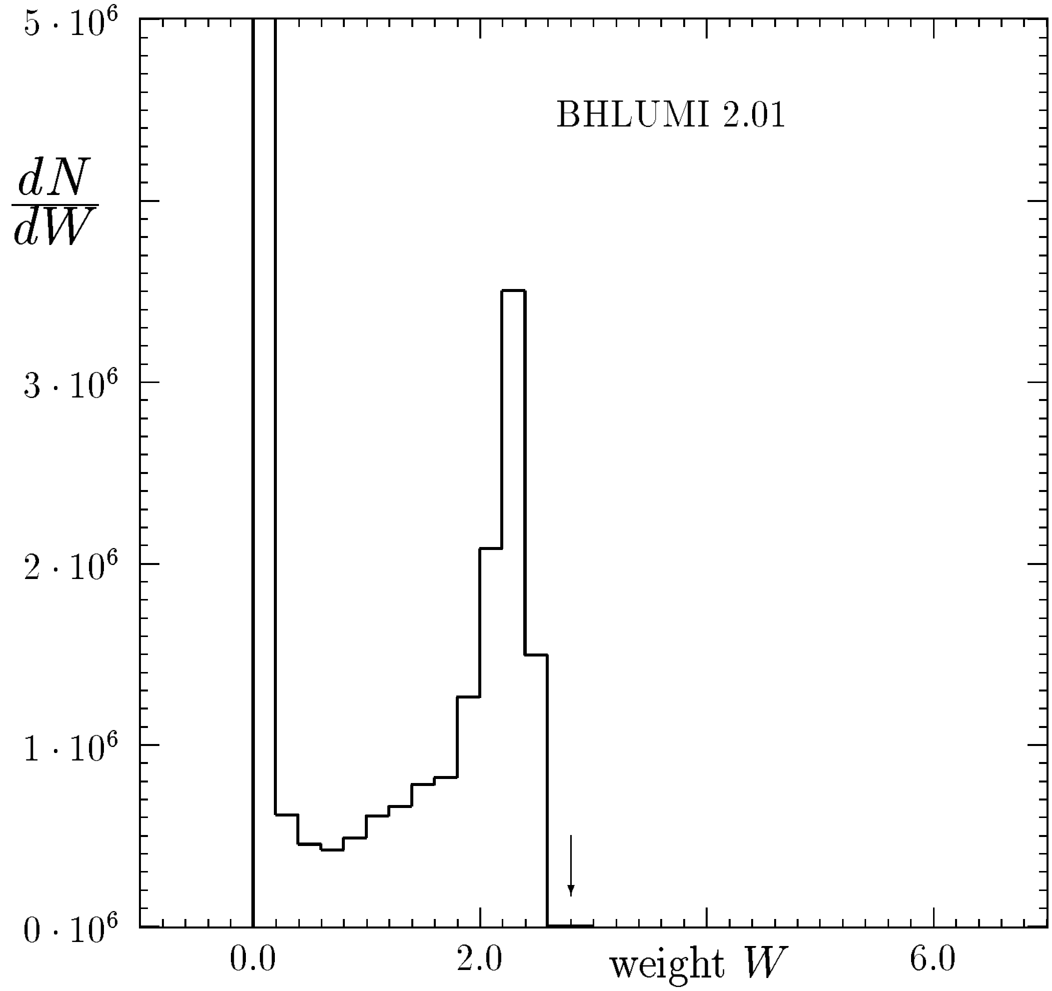

Figure 2: *The distribution of the total weight* $W = W_{crude} W_{model}$ *for multiphoton sub-generator* `BHLUM2`. *Results come from the Monte Carlo sample of* $2.5 \cdot 10^7$ *events. The maximum weight* $W_{\max} = 2.8$ *which is used for (eventual) rejection is marked with arrow. The events with* $W > W_{\max}$*, not visible in the plot, represent* $(5.1 \pm 0.5) \times 10^{-7}$ *of the total cross-section. There is no events with* $W < 0$*. (The highest bin containing events with* $W = 0$ *extends to* $10.8 \cdot 10^6$*.)*

Note that the two exponents cancel precisely due to appropriate choice of Y_Δ. (In eq. (24) there is freedom in the choice of the relative normalization of W and σ_0 which we used in order to get $< W >$ close to one.)

The crude distribution is not only analytically integrable but it is quite simple to generate randomly points $(t, \phi, \tilde{\alpha}_i, \tilde{\beta}_i, \tilde{\alpha}'_i, \phi_i, \tilde{\beta}'_i, \phi'_i)$ according to this distribution. All variables can be, in fact, generated independently! The first four components of the weight which we call together "crude weight"

$$W_{\rm crude} = W^{(1)} W^{(2)} W^{(3)} W^{(4)} \tag{38}$$

are calculated during generation of the phase space variables $(t, \phi, \tilde{\alpha}_i, \tilde{\beta}_i, \tilde{\alpha}'_i, \tilde{\beta}'_i)$ and their translation into final-state four-momenta p_2, q_2, k_i. The last weight to which we usually refer as "model component weight"

$$W_{\rm model} = W^{(5)} \tag{39}$$

is calculated in the separate subprogram. For the purpose of various tests we calculate several variants of the model component weight $W_{\rm model}$, all of them are stored in the common block and are accessible to the user, see Section 4 for more details. This organization, with parallel wights in a single M.C run, allows for *tremendous gains of CPU time* in testing the program – one may calculate in the single Monte Carlo run several cross-sections with different versions of the model component weight (different version of QED matrix element) instead of changing input parameters and running the program several times! This trick is applied most easily in the Monte Carlo run with variable-weight events. The distribution of the total weight $W = W_{\rm crude} W_{\rm model}$ is shown in Fig. 2. It is highly regular, without $W < 0$ events and with totally negligible tail $W > W_{\rm max}$ where $W_{\rm max} = 2.8$ is the maximum weight used in the case of rejection. In fact, the events with $W > W_{\rm max}$ represents only fraction $(5.1 \pm 0.5) \times 10^{-7}$ of the total cross-section.

Why does our Monte Carlo algorithm have so simple a structure, i.e. single rejection loop with single weight? It would definitely be possible to do additional branches in the Monte Carlo algorithm which improve generation efficiency, for example, for extremely hard photons, or to introduce internal rejection loops for a part or the whole of the crude weight. The example of such a sophisticated Monte Carlo optimization can be found in the classic QED event generator MUSTRAAL [28]. We did not do such optimizations in the present program consciously and on purpose. In view of the fact that establishing high *technical precision* of the program costs as much effort as its construction, the clear and clean structure of the algorithm and the program is of highest priority. By optimizing the algorithm we would typically gain perhaps 30% in CPU time per event, which in view of slow convergence of the Monte Carlo integration is *not worth the additional effort* in testing the program.[17]

[17] Especially, if one remembers that the typical speed of the computer's CPU improves at the rate of 20-30% annually.

As indicated above we have implemented in the present version of the program a new method of introducing common soft-photon region $\Omega_{\rm CMS}(\varepsilon)$ for photons generated from electron and positron lines. The region $\Omega_{\rm CMS}(\varepsilon)$ is defined through the $k^0 < \varepsilon\frac{1}{2}\sqrt{s}$ condition in the CMS center-of-mass (laboratory) frame. Implementation of this method is not really necessary for the purpose of the small-angle Bhabha's where up-down interferences are negligible and bremsstrahlung from electron and positron lines can be considered completely independently. This part of the algorithm will be useful, however, for future extension of the program to wide angles. It improves also a little bit the efficiency of the program (but this is not the main aim). The improvement in efficiency is more significant with respect to BHLUMI 1.xx, where we used rather crude method of introducing $\Omega_{\rm CMS}$ by means of assigning zero weight to all events where at least one photon falls into $\Omega_{\rm CMS}$. Another gain is that the following discussion will also explain in more detail the particular choice of our mass weights $W^{(1,2)}$.

To start, let us note that, if soft real photons were generated in the crude distribution ρ_0 according to the exact soft-photon distribution $d\omega_i$, then there would be no problem with transforming two soft-photon regions Ω_1 and Ω_2 into the single region $\Omega_{\rm CMS}(\varepsilon)$ at all! The method would be extremely simple: generate photons with Ω_1 and Ω_2 taking $\Delta << \varepsilon\vartheta^2_{\rm min}$ and *remove* from the list all photons which fall into $\Omega_{\rm CMS}(\varepsilon)$, i.e., which have energy below $\varepsilon\frac{1}{2}\sqrt{s}$ – no need to worry about changing the YFS form factor or anything else in the program. Why would it be possible to proceed like that? It is enough to consider the $\Omega_1 \to \Omega_{\rm CMS}(\varepsilon)$ transition. This transition corresponds to the following change in the YFS form factor (from the real photon part)

$$\exp\left(\int\limits_{\Omega_{\rm CMS}/\Omega_1} d\omega \right), \tag{40}$$

where $\Omega_{\rm CMS}/\Omega_1$ is the (finite) topological difference of the two soft domains. As a short algebraic calculation shows, this factor is, however, compensated exactly by the similar exponential factor due to reduction of the real soft-photon phase space. The Monte Carlo total weight, integrated cross-sections etc., are therefore the same. The above reasoning simply reflects the basic fact that the soft-photon boundary defined by Ω is a dummy parameter in the game and none of the physical results depends on its choice. The only obstacle to using the above prescription is that in our present algorithm we generate very soft photons according to the crude distribution $d\rho_0$, which includes the approximate density $d\tilde{\omega}$ and not the exact density $d\omega$. Are we really forced to do it? At first sight no, because we could have easily modelled, with the additional rejection loop, the exact soft-photon density $d\omega$ in the crude distribution (we could proceed with removing photons as above). We do not do it this way for important reasons. The exact soft-photon distribution $d\omega$ has helicity zeros at the zero photon angle with respect to the fermion. For hard photons these zeros (partly) disappear and the model component weight $W_{\rm model}$ is responsible for filling zero-angle dips in the angular photon distribution. If these zeros were introduced already (by rejection) in the crude distribution $d\rho_0$ then the model component weight would have strong unacceptable fluctuations due to occasional division by (almost) zero. Is there another solution to compromise the requirement of not having helicity zeros in $d\rho_0$ with the nice and easy prescription of removing unwanted soft photons from the list? Yes, and it is the following. In the removal prescription, as described above, soft photons do not contribute to the total weight and its average at all because they are already treated properly in the crude density $d\rho_0$. In the case of our crude distribution present in the program, since the ratio $d\omega/d\tilde{\omega}$ fluctuates for removed soft photons, we have to take this into account, i.e., we have to know analytically and include in the total weight the average $< d\omega/d\tilde{\omega} >$. In fact, it is not difficult to calculate it. Let us note that for the present ρ_0 with $d\tilde{\omega}$'s, the previously described exact compensation between the form factor and the real soft-photon phase space contribution (in $\Omega_1 \to \Omega_{\rm CMS}(\varepsilon)$ transition) is now disturbed. This disturbance corresponds precisely to the average weight $< d\omega/d\tilde{\omega} >$ in which we are interested. In the present case, the form-factor contribution (40) is the same but the factor corresponding to shrinking real soft-photon phase space

$$\exp\left(- \int\limits_{\Omega_{\rm CMS}/\Omega_1} d\tilde{\omega} \right) \tag{41}$$

is different! The product of the two represents the average mass weight of the removed photons for which we are looking

$$\begin{aligned}
\left\langle \prod_{\substack{\rm photons\\ \rm removed}} \frac{d\omega_i}{d\tilde{\omega}_i} \right\rangle &= \exp\left(- \int\limits_{\Omega_{\rm CMS}/\Omega_1} d\tilde{\omega} + \int\limits_{\Omega_{\rm CMS}/\Omega_1} d\omega \right) &(42)\\
&= \exp\left\{ 2\frac{\alpha}{\pi} \left[\ln\left(\frac{\delta_p}{\delta_s}\right) + 1 \right] \ln\frac{\varepsilon_1}{\Delta_p} \right\}, &(43)\\
\varepsilon_1 &= \sqrt{\frac{E_{\rm min}^2}{p^0_{1,CMS}\; p^0_{2,CMS}}}\,.
\end{aligned}$$

The treatment of the mass weights from the positron line is completely analogous. In the actual implementation we control these weights numerically, if the above identity holds, up to the statistical error. The ratio of the left and right side of Eq. (44) is printed in the output window C, positions C4, C5 and C6, of the final printout from sub-generator BHLUM2.

The last subject in this Section is the numerical instability related to the QED matrix element calculation. If we had calculated the model component weight taking the QED matrix element from Eq. (11) and plugging in dot-products of the four-momenta reconstructed from the internal phase space variables $t, \phi, \tilde{\alpha}_i, \tilde{\beta}_i, \phi_i, \tilde{\alpha}'_i, \tilde{\beta}'_i, \phi'_i$, and, next, calculated the total weight by multiplying it by the crude weight, then we would have encountered fluctuations of the total weight at the photon angle of order of δ_p due to rounding errors. We traced most of them to the simple fact that even in 64-bit floating point arithmetic the dot product of two very collinear four-vectors (photon and electron momenta) may easily change by a factor of two or more after these four-vectors are transformed from one Lorentz frame (frame of Monte Carlo generation of photons) to another frame (laboratory frame). The radical solution is to keep track of the original values of the dot product before the Lorentz boost i.e. to express all parts in the QED matrix element which feature collinear zeros/singularities, like $\tilde{S}$ factors and $D_1^{(1)}$ distributions in Eq. (11), directly in terms of the $\tilde{\alpha}$'s and $\tilde{\beta}$'s

$$
\begin{aligned}
&D_1^{(1)}(p_1,p_2,q_1,q_2,k_i)\\
&\quad= \frac{s^2+u^2+s_1^2+u_1^2}{4|t_q|(kp_1)(kp_2)}\left[\frac{\tilde{\alpha}_i\tilde{\beta}_i}{y_i z_i}+\delta_p\frac{y_i^2+z_i^2}{(1-y_i)^2+(1-z_i)^2}\left(\frac{y_i}{z_i}+\frac{z_i}{y_i}\right)\right], \qquad (44)\\
&D_{1'}^{(1)}(p_1,p_2,q_1,q_2,k'_j)\\
&\quad= \frac{s^2+u^2+s_1^2+u_1^2}{4|t_p|(k'_jq_1)(k'_jq_2)}\left[\frac{\tilde{\alpha}'_j\tilde{\beta}'_j}{y'_j z'_j}+\delta_q\frac{{y'_j}^2+{z'_j}^2}{(1-y'_j)^2+(1-z'_j)^2}\left(\frac{y'_j}{z'_j}+\frac{z'_j}{y'_j}\right)\right], \qquad (45)
\end{aligned}
$$

$$
\tilde{S}(p_1,p_2,k_i)=\frac{|t_p|}{(k_ip_1)(k_ip_2)}\,\frac{\tilde{\alpha}_i\tilde{\beta}_i}{y_iz_i},\quad \tilde{S}(q_1,q_2,k_j)=\frac{|t_q|}{(k_jq_1)(k_jq_2)}\,\frac{\tilde{\alpha}'_j\tilde{\beta}'_j}{y'_jz'_j}, \qquad (46)
$$

$$
y_i=\tilde{\alpha}_i+\delta_p\tilde{\beta}_i,\quad z_i=\tilde{\beta}_i+\delta_p\tilde{\alpha}_i,\quad y'_j=\tilde{\alpha}'_j+\delta_q\tilde{\beta}'_j,\quad z'_j=\tilde{\beta}'_j+\delta_q\tilde{\alpha}'_j, \qquad (47)
$$

$$
(k_ip_1)\equiv(p_1\Pi_p)z_i,\quad (k_ip_2)\equiv(p_1\Pi_p)y_i,\quad \Pi_p=p_2+\sum_{i=1,n}k_i, \qquad (48)
$$

$$
(k'_jq_1)\equiv(q_1\Pi_q)z'_j,\quad (k'_jq_2)\equiv(q_1\Pi_q)y'_j,\quad \Pi_q=q_2+\sum_{j=1,n'}k'_j. \qquad (49)
$$

The above expressions are equivalent to the original expressions of Eq. (11) up to terms of $\mathcal{O}(m_e^2/|t|)$.

2.3 LUMLOG Monte carlo algorithm

The leading-logarithmic (LL) sub-generator LUMLOG is closely related to Ref. [3] where the reader may find numerical results from this program and detailed discussion of the physics which they represent. This Monte Carlo event sub-generator works in the strict collinear approximation for the initial and final state photon emission. In contrast to the other two sub-generators it provides only variable-weight events. It should be always remembered that final state four-momenta represent "dressed" $e^{\pm}$, i.e. normal "bare" $e^{\pm}$ together with all accompanying bremsstrahlung photons. This is dictated by the calorimetric character of the typical LEP/SLC luminosity detector. The program is able to calculate in a single Monte Carlo run up to eight QED leading-logarithmic cross-sections/corrections, in various perturbative orders, up $\mathcal{O}(\beta_t^3)$ in the LL expansion parameter β_t, see Section 4. We use the following two definitions of this effective LL coupling constant[18]

$$\beta_t^{(A)} = \frac{\alpha}{\pi}\Big(\ln\frac{|t|}{m_e^2}-1\Big),\quad \beta_t^{(B)} = \frac{\alpha}{\pi}\ln\frac{s\xi_{\min}}{m_e^2} = \frac{\alpha}{\pi}\ln\frac{|t_{\min}|}{m_e^2}. \tag{50}$$

In the case of $\beta_t^{(A)}$ the $|t|$ denotes the true four-momentum transfer exchanged in the hard scattering between the two electron lines. Both choices are legal LL choices but we prefer $\beta_t^{(A)}$ because it minimizes the sub-leading contribution in the soft photon limit. Let us also remark that whenever we use "exponentiation" in the context of the LL calculations, then we do not mean the full YFS exponentiation but rather its Gribov-type restriction to the LL context, see Refs. [29, 25, 30, 10].

This sub-generator was thoroughly tested by comparisons with another Monte Carlo program MULTILOG and semi-analytical calculations for "artificial" testing of the electron structure function in a wide range of input parameters. From the agreement of these calculations we infer the *technical precision* of LUMLOG to be 0.02% in the total cross-section. The similar LL calculations were done in Ref. [31] but the authors do not quote the technical precision separately and claim the overall precision (including higher orders beyond second order) to be 0.5%.

The master formula for the small-angle Bhabha total cross-section with the initial-state bremsstrahlung in the LL approximation reads

$$\begin{aligned}\sigma^{\rm LL} &= \int_{x_{\min}}^{1} dx_1 \int_{x_{\min}}^{1} dx_2 \int_0^1 d\xi^* \int_0^{2\pi} d\phi\; D(x_1,\beta_t)D(x_2,\beta_t)\\ &\qquad \frac{d\sigma^{\rm Born}}{d\phi d\xi^*}\;\Theta_{\xi_a}^{\xi_b}(\xi_1)\;\Theta_{\xi_a}^{\xi_b}(\xi_2),\end{aligned} \tag{51}$$

where ϕ is the azimuthal angle around the beam, x_1, x_2 denote fractions of the energies carried by the beams, after emission of the initial-state photons. The energy cut in the typical experiment is roughly $x_{\min} \simeq 0.5$ while in the program it may be set $x_{\min} = 10^{-4}$ and lower.[19] Any other energy cut can be imposed by rejection.[20] The angular trigger as defined by

$$\Theta_{\xi_a}^{\xi_b}(\xi) = \Theta(\xi-\xi_a)\Theta(\xi_b-\xi) \tag{52}$$

is symmetric, but since we use the Monte Carlo integration/simulation method, it will be trivial to turn it into an asymmetric or any other type of (calorimetric)

---

[18]The introduction of light $e^{\pm}$ pairs in the LL approximation is done by the introduction of the running QED coupling constant into the structure functions via their evolution equations [16, 11]; this amounts to the substitution $\beta_t^{(A)} \to -3\ln\big(1-(\alpha/3\pi)(\ln(|t|/m_e^2)-1)\big)$ and is equivalent to adding real/virtual electron pairs to a non-singlet structure function in the LL approximation.

[19]Simple Monte Carlo exercise shows that to a very high precision $x_i = E_i^{\rm cluster}/E_{\rm beam}$ where $E_i^{\rm cluster}$ is the calorimetric energy of outgoing dressed $e^{\pm}$. Variables x_i are therefore directly measurable!

[20]In Ref. [3] we have used $1-s'/s = 1-x_1x_2 = v < v_{\max}$ cut.

trigger. The Born cross-section with pure t-channel photon exchange reads

$$\frac{d\sigma^{\rm Born}}{d\phi d\xi^*} = \frac{\alpha^2}{s x_1 x_2}\frac{1+(1-\xi^*)^2}{2\xi^{*2}}, \tag{53}$$

where ϑ^* in $\xi^* = (1-\cos\vartheta^*)/2$ is defined as the scattering angle in the LL hard scattering rest frame[21] and $\xi_{1,2} = (1-\cos\vartheta_{1,2})/2$ are convenient parametrization of the laboratory scattering angles $\vartheta_{1,2}$ of $e^\pm$. The $\xi_{1,2}$ are related to ξ^* as follows

$$\xi_1 = \frac{x_2\xi^*}{x_2\xi^* + x_1(1-\xi^*)}, \qquad \xi_2 = \frac{x_1\xi^*}{x_1\xi^* + x_2(1-\xi^*)}. \tag{54}$$

All the above was kinematics and the QED perturbative LL calculation is located in the so-called electron structure functions $D(x,\beta)$, which in QED can be calculated with an arbitrary precision. In our calculations we shall mainly use the non-singlet (valence) structure functions calculated perturbatively in the $\mathcal{O}(\beta_t^1)$, $\mathcal{O}(\beta_t^2)$ and exact $\mathcal{O}(\beta_t^\infty)$, see Refs. [16, 30, 10] for more discussion. We use in the program the second-order non-exponentiated electron structure function (convolution) which reads as follows

$$\begin{aligned}
D(x_1,\beta_t)D(x_2,\beta_t)|_{\mathcal{O}(\beta_t^2)} &= \delta(1-x_1)\delta(1-x_2)(1+\Delta)\\
&+\delta(1-x_1)\;\theta(1-x_2-\epsilon)f_1(x_2) + \delta(1-x_2)\;\theta(1-x_1-\epsilon)f_1(x_1)\\
&+\frac{1}{4}\theta(1-x_1-\epsilon)\;\theta(1-x_2-\epsilon)\;\beta_t^2\;\frac{1+x_1^2}{1-x_1}\;\frac{1+x_2^2}{1-x_2},
\end{aligned} \tag{55}$$

$$\Delta = \beta_t\;(\frac{3}{2}+2\ln\epsilon) + \frac{1}{4}\beta_t^2\;(\frac{3}{2}+2\ln\epsilon)^2 + \frac{1}{2}\beta_t^2\;\Big(2\ln^2\epsilon + 3\ln\epsilon + \frac{9}{8} - \frac{\pi^2}{3}\Big),$$

$$\begin{aligned}
f_1(x) = \frac{1}{2}\;\frac{1+x^2}{1-x}\;&\Big\{\beta_t + \frac{1}{2}\beta_t^2\Big(\frac{3}{2}+2\ln\epsilon\Big) + \frac{1}{2}\beta_t^2\Big(2\ln(1-x) - \ln x + \frac{3}{2}\Big)\Big\}\\
&+\frac{1}{4}\beta_t^2\;\Big(\frac{1}{2}(1+x)\ln x - 1 + x\Big)
\end{aligned}$$

and the $\mathcal{O}(\beta_t^1)$ and Born $\mathcal{O}(\beta_t^0)$ truncations as well. We recommend, however, using the YFS-exponentiated (in the Gribov sense) expressions for electron structure functions which are much closer to the exact infinite-order solution of the evolution equations. The best available exponentiated expression for the non-singlet structure function in $\mathcal{O}(\beta_t^3)$ comes from Ref. [10] and is indistinguishable, at our technical precision level, from the exact $\mathcal{O}(\beta_t^\infty)$ solution of the non-singlet evolution equation [30]. It reads as follows

$$D_3^{\rm YFS}(x,\beta_t) = \frac{\exp\left(\beta_t(\frac{3}{4}-C)\right)}{\Gamma(1+\beta_t)}\;\beta_t(1-x)^{\beta_t-1}\Big\{1-\frac{1}{2}(1-x^2)$$

[21] In small-angle Bhabha's ϑ^* is practically directly measurable because it can be trivially obtained from the laboratory (dressed) electron scattering angles ϑ_i through the approximate relation $\vartheta^* = \sqrt{\vartheta_1\vartheta_2}$.

$$+\beta_t\left[-\frac{1}{8}(1+3x^2)\ln(x)-\frac{1}{4}(1-x)^2\right]+\beta_t^2\left[\frac{1}{8}(1-x)^2\right. \tag{56}$$
$$\left.\left.+\frac{1}{16}(3x^2-4x+1)\ln(x)+\frac{1}{96}(1+7x^2)\ln^2(x)+\frac{1}{8}(1-x^2)\ \mathrm{Li}_2(1-x)\right]\right\},$$

$C=0.577215...$, and the program provides its $\mathcal{O}(\beta_t^2)$, $\mathcal{O}(\beta_t^1)$ and $\mathcal{O}(\beta_t^0)$ exponentiated versions as well.[22]

In the phase space integration of Eq. (51) we have four variables, x_1, x_3, ξ^* and ϕ which determine completely the final-state four-momenta of $e^{\pm}$. The main singularities in $d\rho^{LL}$ are: infrared singularity $(1-x_i)^{\beta_t-1}$, t-channel singularity $1/\xi^{*2}$ and kinematical singularity $1/(x_1x_2)^2$. In the first step towards the Monte Carlo algorithm we re-arrange integration variables. We eliminate ξ^* at the expense of one of ξ_1 or ξ_2. In order to preserve symmetry of the algorithm and to improve treatment of the angular constraints due to Θ's we do it as follows: If $x_1 > x_2$ we substitute ξ^* by ξ_1, else we substitute ξ^* by ξ_2. Let us consider the first case

$$\sigma^{\mathrm{LL}} = \frac{\alpha^2}{s}\int_0^{2\pi} d\phi \int_{\xi_a}^{\xi_b}\frac{d\xi_1}{\xi_1^2}\int_{x_{\min}}^{1} dx_1 \int_{x_{\min}}^{1} dx_2 \frac{\theta(x_1-x_2)}{x_1^2}\,\frac{1}{2}\left[1+\left(1-\frac{x_1\xi_1}{x_2+\xi_1(x_1-x_2)}\right)^2\right]$$
$$\times D(x_1,\beta_t)D(x_2,\beta_t)\ \Theta_{\xi_a}^{\xi_b}\left(\frac{x_1^2\xi_1}{x_2^2+\xi_1(x_1^2-x_2^2)}\right)=\int d\rho^{\mathrm{LL}}. \tag{57}$$

Next, we simplify the integrand $d\rho^{\mathrm{LL}} \to d\rho_0^{\mathrm{LL}}$, countering the simplification with the weight $W = d\rho^{\mathrm{LL}}/d\rho_0^{\mathrm{LL}}$. The crude distribution $d\rho_0^{\mathrm{LL}}$ we define as follows

$$\sigma_0^{\mathrm{LL}} = \frac{\alpha^2}{s}\int_0^{2\pi} d\phi \int_{\xi_a}^{\xi_b}\frac{d\xi_1}{\xi_1^2}\int_{x_{\min}}^{1} dx_1 dx_2 \frac{\theta(x_1-x_2)}{x_1^2}\ D_0(x_1,\beta_t)D_0(x_2,\beta_t)\ \theta\left(\frac{x_2}{x_1}-\frac{\xi_a}{\xi_b}\right)$$
$$=\int d\rho_0^{\mathrm{LL}}, \tag{58}$$

where

$$D_0(x,\beta)=\beta(1-x)^{\beta-1}\ \frac{1+x^2}{2} \tag{59}$$

is the simplified structure function. Now, variables ϕ and ξ_1 are generated independently and the double differential distribution in variables x_1 and x_2 is generated separately with the help of the universal two-dimensional Monte Carlo sampler VESK2W, which is able to generate arbitrary two-dimensional distribution over the $0 \le x_{1,2} \le 1$ square. This program was already used in LESKOF [24] and has the advantage of being very stable numerically and providing a solid estimate of the numerical/statistical error of the calculated integral. The total cross-section is given by

$$\sigma^{\mathrm{LL}} = \langle W\rangle \int d\rho_0^{\mathrm{LL}} = \langle W\rangle\,\sigma_0^{\mathrm{LL}}, \tag{60}$$

[22] The superscript YFS means that the LL exponentiation is closely related to full YFS exponentiation, see Refs. [20, 10, 32].

where $< W >$ is the average weight and $\sigma_0^{\rm LL}$ is provided by VESK2W.

The Monte Carlo algorithm is (obviously) constructed for the exponentiated structure function of Eq. (57), which is smooth at the $x_1 \to 1$ soft limit. The non--exponentiated structure functions of Eq. (56) have in this limit, from the Monte Carlo point of view, the acute δ-like singularities. Do we have to construct a separate Monte Carlo program for them? No, there is another way out, and the method of implementing exponentiated and non-exponentiated QED calculations in the single Monte Carlo program is technically very important for future developments and tests of the QED Monte Carlo event generators. How do we proceed, for example, with the product of $\mathcal{O}(\beta_t^0)$ structure functions, $D(x_1,\beta_t)D(x_2,\beta_t) = \delta(x_1)\delta_(x_2)$ (which simply represents the Born calculation)? The crude distribution $D_0(x_1,\beta_t)D_0(x_2,\beta_t) = \beta_t(1-x_1)^{\beta_t-1}\beta_t(1-x_2)^{\beta_t-1}$ for typical $\beta_t \simeq 0.03$ is not, in fact, very much different from the previous product of δ's. The Monte Carlo procedure is the following: we assign the *constant*, properly normalized model component weight to events with $\max(1-x_1, 1-x_2) < \epsilon = 10^{-5}$ generated according to crude distributions $\beta_t(1-x_1)^{\beta_t-1}\beta_t(1-x_2)^{\beta_t-1}$. They play the role of events generated according to the product $\delta(x_1)\delta(x_2)$! The events with $\max(1-x_1, 1-x_2) > \epsilon = 10^{-5}$ are assigned the model component weight equal zero. In the case of the full $\mathcal{O}(\beta_t^2)$ expression (56) we proceed in the analogous way, we split the $0 < x_{1,2} < 1$ square into four sectors: (a) $1-x_1 < \epsilon,\;\; 1-x_2 < \epsilon$, (b) $1-x_1 < \epsilon,\;\; 1-x_2 > \epsilon$, (c) $1-x_1 > \epsilon,\;\; 1-x_2 < \epsilon$, (d) $1-x_1 > \epsilon,\;\; 1-x_2 > \epsilon$. Events from these sectors we identify with four terms in Eq. (56) which are proportional to $\delta(x_1)\delta(x_2)$, $\delta(x_1)$, $\delta(x_1)$ and the no-delta term. The whole art is in constructing a properly normalized model component weight for δ-like terms. This method is definitely a bit inefficient with respect to direct generation of δ's but the other advantages are far more important. First of all, there is no need to write and test another new generator based on another algorithm – our solution amounts merely to writing another subroutine for the model component weight. Second, the typical use of LUMLOG is the calculation of the difference $\mathcal{O}(\beta_t^3)_{exp} - \mathcal{O}(\beta_t)$, which would require two Monte Carlo runs of separate programs, while in our case we have the result from the single Monte Carlo run (this outweighs any inefficiency loss with respect to the two-program solution). Let us note that a very similar method was already used in BHLUMI 1.xx, where the non-exponentiated QED $\mathcal{O}(\alpha)$ model component weight was imposed on the top of the crude exponentiated multi-photon distribution. In future developments of the actual version of BHLUMI it will be possible to implement the entire LUMLOG leading-logarithmic calculation as another model component weight in the multiphoton sub-generator BHLUM2. The comparison with results of the original LUMLOG will provide a powerful test of the technical precision of BHLUMI.

2.4 Modifications of OLDBIS with respect to OLDBAB

We have started with the source code of OLDBAB taken from the RADCOR package installed on the CERN public disk by R. Kleiss. Most of the source code in the present OLDBIS is identical to that of OLDBAB. The Monte Carlo algorithm is almost unchanged. New corrections are of two types: (a) cosmetic ones which concern mainly output/input formats, choice of random number generator, etc. (b) essential ones which modify the algorithm, implementation of the QED matrix element and usage of program.

Let us list, first, the less important cosmetic changes

- The histogramming routines are removed.
- The input/output structure was aligned with other Monte Carlo programs like KORALZ, LESKOF etc. In the `CALL OLDBIS(MODE,XPAR,NPAR)` initialization, generation and postgeneration phases are determined by the `MODE= -1,0,1` parameter.
- All input is read through XPAR and NPAR in the obligatory initialization for `MODE=-1` and immediately printed out.
- Each generated event (for `MODE=0`) is encoded in `/MOMSET/` and `/WGTALL/` common blocks.
- Calling OLDBIS with `MODE=1,2` provides the cross-sections and other auxiliary information resulting from the Monte Carlo integration (necessary for histogram normalization) through `NPAR and XPAR` parameters. For `MODE=1` no output is printed.
- For `MODE=2`, in addition to information encoded in `XPAR` and `NPAR`, certain output is printed. Note that the routine `BABINF`, printing output in the same format as in the original OLDBAB, is kept active.
- All weight accounting is not done "by hand" anymore, but rather by using the special routine `WMONIT`.
- The modern random number generators RANMAR and RANECU replace the original one.
- Output four-momenta `QP,QM,QK` are, now, in GeV units.
- The values of physical constants like α are not hard-wired directly into formulas but are defined as the double precision variables.

The more important changes concerning the Monte Carlo algorithm and QED matrix element are the following

- Detailed insight into Monte Carlo algorithm of OLDBAB reveals that the variable `SIGS` is a dummy variable, i.e., none of the calculated cross-sections and distributions depends (within statistical error) on `SIGS`. For this to be true `SIGS` has to be positive, however! For very small k_0 = `XKO` the variable

SIGS becomes negative and the results from the OLDBAB do not represent the true first order QED anymore. We have corrected this, i.e., defined a SIGS which is always positive.

- The above redefinition of SIGS is still not enough to avoid problems at the very small k_0 = XK0 limit, which has to be taken in order to check that the program represents the first order QED at the 0.02% technical precision level. One has to allow for negative weights, i.e., the introduction of weighted events is necessary. This option is implemented and the program provides weighted events for the input parameter switch KEYWGT=1. The total weight WTM is located in /MOMBAB/ and later transferred to /WGTALL/ as WTMOD. We recommend using OLDBIS with weighted events.

- The essential observation made in Ref. [4], and exploited also in other papers [3, 2], is that a certain class of annoying QED corrections, so called up-down interferences, is completely unimportant at small-angles ($< 10^o$). To obtain this result we had to rewrite completely the QED soft and hard matrix element in OLDBAB. The new routine VIRSEL calculating soft and virtual corrections is added. The user has at his disposal four types of matrix element with various components switched on/off. Each of them is represented by the separate model component weight XWT(10:13), see Table 4. Only one of them XWT(10+KEYSIN) is chosen as a model component weight in the total weight. The difference XWT(12)-XWT(11) accounts for the pure up-down interference. Certain backward compatibility is kept – XWT(10) represents the original OLDBAB matrix element, but with new vacuum polarization and Z width.

- The archaic vacuum polarization subprogram REPI is replaced by the modern version of the same name [33].

- The Z-γ interference correction includes now the Z width.

- There is, now, no immediate rejection for events in which one of the outgoing $e^{\pm}$'s is in the angular trigger but another one (due to photon emission) is out. Such an event passes through, but has weight equal zero.

- Symetrization $e^- \leftrightarrow e^+$ can be suppressed by setting the special switch KEYMIR = 1. This option was useful in comparisons of OLDBIS results in Ref. [4] with the semi-analytical calculations.

All these changes are marked in the source code with special string of characters for easy identification.

3 Subprograms in the generator

The present version of the Monte Carlo generator BHLUMI includes three independent sub-generators BHLUM2, LUMLOG and OLDBIS which form three distinct parts of the code. They are supplemented with the library of the utility subprograms named BHLLIB. The main program and two demonstration programs BHLDE1 and BHLDE2 are described in the next Section. The only purpose of subroutine `BHLUMI` is to call on one of three sub-generators and to pass the arguments. In fact, each of the three sub-generators may be called directly and it is possible to separate it completely from the rest of the program rather easily.

The important subprograms in BHLUM2 part are the following

- `BHLUM2` is the central administrative subprogram. It calls on all other subprograms performing essential steps in the Monte Carlo event generation. It controls weight book-keeping, flow of input, output and communication among other subprograms in the entire BHLUM2 sub-generator.

- `FILBHL2` in the initialization stage defines variables in the common blocks using information from `XPAR` and `NPAR`. It also prints useful output related to input parameters.

- `MLTIBR` is the most important subprogram in the generator. It generates Sudakov-type variables $\tilde{\alpha}_i$, $\tilde{\beta}_i$, azimuthal angles ϕ_i and calculates four momenta of photons and fermions. All this is done for one fermion line in the t channel rest frame QRS_p or QRS_q. It is called twice, separately for e^- and e^+ lines. It also calculates components for the mass weights.

- `PIATEK` is the subprogram which calculates two versions of the mass-weights $W^{(1)}$ and $W^{(2)}$: one version for the case where soft photon removal is not done and a second version for the case of removal. In the second case it also calculates the relevant control weight. `REMPHO` does the actual job of removing soft photons with the energy below CMS cut $\varepsilon\sqrt{s}/2$ (in the case where this version of the algorithm is actually applied).

- `POISSG` generates photon multiplicity `NPHOT`.

- `KINO4` transforms four-momenta of $e^{\pm}$ and photons from QRS_p and QRS_q to the CMS laboratory system.

- `MERGIK` puts four-momenta of all photons in the single common block ordering them according to CMS energy.

- `MODEL1` calculates the list of the model component weights.

- `REPI` calculates the total (leptonic plus hadronic) photon vacuum polarization[23] according to the parametrization of Ref. [33].

[23]We are grateful to Dr. H. Burkhardt for providing us with this subprogram.

- DUMPS prints final state four-momenta – useful for test.

The important subprograms in the LUMLOG part are the following

- LUMLOG transfers call to subroutine BHALOG.
- BHALOG is the central administrative subprogram. It calls on all other subprograms performing essential steps in the Monte Carlo event generation. It controls weight book-keeping, flow of input, output and communication among other subprograms.
- MODELU calculates the model component weight for $\mathcal{O}(\beta_t^2)$ non-exponentiated structure functions.
- KINOLT constructs four-momenta out of phase space variables ϕ, ξ_1, x_1, x_2.
- FUNSKI provides the two dimensional crude distribution in x_1, x_2 to be generated by VESK2W.
- STRUFU calculates the exponentiated singlet (valence) structure function of the electron, four versions from $\mathcal{O}(\beta^0)$ to $\mathcal{O}(\eta^3)$.

The important subprograms in the OLDBIS part are the following

- OLDBIS includes all of the generation Monte Carlo algorithm including weight book-keeping and flow of input/output, printout of final results, etc.
- BABINF old routine which prints final results on the total cross-sections in the same format as in OLDBAB.
- VIRSEL is the new routine calculating various versions of the virtual corrections – notably with and without up-down interferences.
- VIRSOF is the old routine calculating virtual corrections.
- YYWEAK is the new version of the function XSWEAK calculating Z exchange correction.
- KINBIS fills final four-momenta into standard common block /MOMSET/ of BHLUMI.

The important subprograms in the utility library BHLLIB are the following

- VARRAN Switchable random number generator. We installed in the program two random number generators RANMAR [34] and RANECU[35].[24] Only one of them is used in a given Monte Carlo run depending on the switch KEYRND, see description of the input data in the next Section. The single routine VARRAN is introduced in order to facilitate the optional use of one of the two generators.

[24]We would like to thank Dr. F. James for providing us the code of the two programs.

- MARRAN is a copy of the standard CERN random number generator RANMAR of Ref. [34] called in VARRAN.

- RANECU is the random number generator of Ref. [35] called through VARRAN.

- VESK2W is the two-dimensional Monte Carlo sampler capable of generating an arbitrary two-dimensional distribution, used also in LESKOF Monte Carlo [24].

- WMONIT monitors up to 100 weights, calculating average, dispersion and maximum of the monitored weight.

- WMONI2 is a copy of WMONIT.

4 How to use the program

In order to start quickly without reading all of this section we propose the following short code:

```
      COMMON / MOMSET / P1(4),Q1(4),P2(4),Q2(4),PHOT(100,4),NPHOT
      DOUBLE PRECISION  P1,P2,Q1,Q2,PHOT,XPAR(100)
      INTEGER NPAR(100)
      (book histograms)
      NPAR(1)= 3001
      NPAR(2)=    1
      XPAR(1)=   92D0
      XPAR(2)=   1.9D0
      XPAR(3)=   64.D0
      XPAR(4)=   1D-4
      CALL BHLUMI(-1,XPAR,NPAR)
      DO 500 IEVENT = 1,10000
      CALL BHLUMI( 0,XPAR,NPAR)
      (fill histograms)
  500 CONTINUE
      CALL BHLUMI( 2,XPAR,NPAR)
      (print histograms)
```

It can be found in the beginning of the demonstration deck and its output starts the test run output at the end of the present paper. Input option switches in NPAR assure that your Monte Carlo events will be produced according to $\mathcal{O}(\alpha)$ Yennie-Frautschi-Suura exponentiation, including Z and s-channel resonance exchange, s-channel γ exchange and t-channel photon vacuum polarization. Events will have constant weight WTMOD=1. Parameter XPAR(1) is CMS energy (GeV), XPAR(2) and XPAR(3) are minimum and maximum t-channel transfers (GeV^2) corresponding approximately to $\vartheta_{\min} = 1.7^\circ$ and $\vartheta_{\max} = 9.9^\circ$. Remember that this generator should be used for $\vartheta_{\mathrm{min,max}} \leq 10°$. The final state $e^{\pm}$ four momenta (GeV) are encoded in P2, Q2 and that of the i-th bremsstrahlung photon in PHOT(i,*). Variable NPHOT is the number of photons. Beam momenta are also provided as P1, Q1. The last call on BHLUMI prints final Monte Carlo statistics, cross-sections etc.

In the general case a complete sequence of calls which generates 10000 events may look like

```
      COMMON / MOMSET / P1(4),Q1(4),P2(4),Q2(4),PHOT(100,4),NPHOT
      COMMON / WGTALL / WTMOD,WTCRU1,WTCRU2,WTSET(100)
      DOUBLE PRECISION  P1,P2,Q1,Q2,PHOT,XPAR(100)
      DOUBLE PRECISION  WTMOD,WTCRU1,WTCRU2,WTSET
      INTEGER NPAR(100)
      define NPAR and XPAR, book histograms
      CALL BHLUMI(-1,XPAR,NPAR)
      DO 500 IEVENT = 1,10000
      CALL BHLUMI( 0,XPAR,NPAR)
      fill histograms
  500 CONTINUE
      CALL BHLUMI( 1,XPAR,NPAR)
      normalize and print histograms
      CALL BHLUMI( 2,XPAR,NPAR)
```

As is obvious from the above example the first integer parameter MODE in any CALL BHLUMI(MODE,XPAR,NPAR) simply decides whether BHLUMI is called in initialization mode (MODE = -1), generation mode (MODE = 0) or post-generation mode (MODE = 1,2). The first obligatory call, with MODE=-1, is employed in order to transfer the input data through NPAR and XPAR parameters and to execute initializations in certain subprograms (no event generation). The series of calls with MODE=0 does the main job, i.e., generate Monte Carlo events. Four-momenta of the final state are accessible in the appropriate common blocks. The last two optional calls with MODE=1,2 can provide useful output information through NPAR and XPAR and also in the form of printout (for MODE=2 only).

Let us now give more details about the input/output parameters in every particular mode.

4.1 Input in initialization mode

The input data are transferred to BHLUMI through XPAR and NPAR parameters in the initialization mode, for MODE=-1. The meaning of the entries NPAR(1) and XPAR(1) is the same for all three sub-generators but this is not true for other entries. We therefore describe the meaning of XPAR and NPAR in three Tables 1-3, separately for the BHLUM2, LUMLOG and OLDBIS sub-generators. The choice of sub-generator is made according to the KEYGEN=1,2,3 switch, see description of NPAR(1).

Parameter	Meaning
NPAR(1)=KEYOPT	=1000*KEYGEN+10*KEYWGT+KEYRND, general option switch, where KEYGEN=3 for this sub-generator, KEYRND=1,2 implies use of RANMAR or RANECU random number generator. If KEYWGT=0 then constant total weight WTMOD=1 events are produced else, for KEYWGT=1 variable weight (weighted) events are provided. In the latter case the user may also exploit weights other than WTMOD, see common block /WGTALL/
NPAR(2)=KEYRAD	= KEYPIA, option switch determining the type of QED matrix element used to calculate the principal total weight WTMOD. Normally the user should use KEYPIA=1. For KEYPIA=0 the t-channel photon vacuum polarization, Z and s-channel γ contributions are all switched off (pure QED bremsstrahlung).
XPAR(1)=CMSENE	$\sqrt{s}$, center-of-mass (CMS) energy in GeV units
XPAR(2)=TRMIN	Minimum t-channel transfer $\|t_{\rm min}\|$ in GeV^2 units
XPAR(3)=TRMAX	Maximum t-channel transfer $\|t_{\rm max}\|$ in GeV^2 units
XPAR(4)=EPSCM	ε, dimensionless infrared cut on CMS energy of soft real photons, $E_{\rm phot} > \varepsilon\sqrt{s}/2$, recommended range $10^{-7} < \varepsilon < 10^{-4}$

Table 1: *List of input parameters of BHLUM2 sub-generator.*

Other parameters, like electron mass $m_e = 0.0005111$ GeV and QED coupling constant $\alpha = 1/137.03604$, are defined inside the program.

4.2 Output in event generation mode

In the event generation mode, for **MODE=0**, both **XPAR** and **NPAR** are ignored and the single Monte Carlo event accompanied with weights is provided to the user in

```
COMMON / MOMSET / P1(4),Q1(4),P2(4),Q2(4),PHOT(100,4),NPHOT
COMMON / WGTALL / WTMOD,WTCRU1,WTCRU2,WTSET(100)
```

Matrices **P1** and **P2** represent four-momenta of incoming (beam) and outgoing e^- while **Q1** and **Q2** represent four-momenta of incoming (beam) and outgoing e^+. List of n =**NPHOT** photon four-momenta is encoded in **PHOT**. For constant weight events, obtained by setting **KEYWGT=0**, we have **WTMOD=1D0** and the entire **/WGTALL/** may and should be ignored. For variable weight, **KEYWGT=1**, events there are two levels of weights. For unsophisticated use of the program one takes events with the total weight **WTMOD** and the type of QED matrix element in **WTMOD** is determined by **KEYRAD=NPAR(2)**. (N.B. for weighted events, this weight is used for the rejection.) The advanced user of the program may, *in the same Monte Carlo run*, use a variety of alternative weights defined as **WTALT= WTCRU1*WTCRU2*WTSET(J)** where each **J** corresponds to one of many versions of the QED matrix element, in various perturbative orders and with various contributions switched on/off. This is explained in

Parameter	Meaning
NPAR(1)=KEYOPT	=1000*KEYGEN+10*KEYWGT+KEYRND, general option switch, where KEYGEN=2 for this sub-generator, KEYRND=1,2 implies use of RANMAR or RANECU random number generator. Only the KEYWGT=1 option of variable weight events is implemented!
NPAR(2)=KEYRAD	= 10*KEYTES +KEYBLO, option switch determining the type of QED matrix element used to calculate the principal total weight WTMOD. The user may exploit weights other than WTMOD, see common block /WGTALL/, described in separate table. Normally KEYTES=0 while KEYTES=1 is for tests only – the QED electron structure functions replaced with (unrealistic) testing functions $(1-z)^{1/2}$. The KEYBLO switch controls the definition of the big logarithm L in the calculation: for KEYBLO=3 we use $L = \ln(s'\xi^\star/m_e^2) - 1$ (recommended choice with proper soft limit) and for KEYBLO=4 we use $L = \ln(s\xi_a/m_e^2)$ (more crude but legitimate choice).
XPAR(1)=CMSENE	$\sqrt{s}$, center-of-mass (CMS) energy in GeV units
XPAR(2)=TMINL	Minimum ϑ electron/positron scattering angle in degree units
XPAR(3)=TMAXL	Maximum ϑ electron/positron scattering angle in degree units
XPAR(4)=XK0	k_0, dimensionless infrared cut on CMS energy of soft real photons, $E_{\rm phot} > k_0\sqrt{s}/2$ for *not-exponentiated* versions of the calculation. Recommended range $10^{-7} < k_0 < 10^{-4}$
XPAR(5)=XKMAX	$k_{\rm max}$, this parameters determines minimum effective mass s' of the final state electron and positron: $s' > s(1-k_{\rm max})$. Note that $k_{\rm max} = 1$ is allowed but we recommend $k_{\rm max} \leq 0.9999$.

Table 2: *List of input parameters of LUMLOG sub-generator.*

Parameter	Meaning
NPAR(1)=KEYOPT	=1000*KEYGEN+10*KEYWGT+KEYRND, general option switch, where KEYGEN=1 for this sub-generator, KEYRND=1,2 implies use of RANMAR or RANECU random number generator. If KEYWGT=0 then constant weight WTMOD=1 events are produced, else for KEYWGT=1 variable weight (weighted) events are provided. In the latter case the user may also exploit weights other than WTMOD, see common block /WGTALL/. For high technical precision we recommend KEYWGT=1.
NPAR(2)=KEYRAD	= 10*KEYMIR+KEYSIN, option switch determining the type of QED matrix element used to calculate the principal total weight WTMOD. KEYSIN=0,1,2,3 are allowed and they represent various options with the up-down interference, vacuum polarization, Z resonance exchange and s-channel photon contributions switched on and off. For KEYSIN=1 all above contributions are switched off. See description of the common block /WGTALL/ for more details. Normally, one should set KEYMIR=0 while KEYMIR=1 is for special tests only [4] – photon emitted only from one fermion line (positron of QP four-momentum).
XPAR(1)=CMSENE	$\sqrt{s}$, center-of-mass (CMS) energy in GeV units
XPAR(2)=TMINL	Minimum ϑ electron/positron scattering angle in degree units
XPAR(3)=TMAXL	Maximum ϑ electron/positron scattering angle in degree units
XPAR(4)=XK0	k_0, dimensionless infrared cut on CMS energy of soft real photons, $E_{\rm phot} > k_0\sqrt{s}/2$. Recommended range $10^{-7} < k_0 < 10^{-4}$ for KEYWGT=1 and $k_0 \simeq 10^{-3}$ for KEYWGT=0.
XPAR(5)=XKMAX	$k_{\rm max}$, this parameter determines maximum CMS energy of the real photon $E_{photon} < k_{\rm max}$. Note that $k_{\rm max} = 1$ is allowed and recommended.
XPAR(6)=XKMIN	$k_{\rm min}$, this parameter determines minimum CMS energy of the real photon $E_{\rm phot} > k_{\rm min}$. Normally $k_{\rm min} = 0$, non-zero values for special tests only

Table 3: *List of input parameters of OLDBIS sub-generator.*

BHLUM2				
Entry	Type of QED calculation			
`WTSET(1)`	`=WTSET(11)`			
`WTSET(2)`	`=WTSET(12)`			
	QED order	Vac. pol.	Z-exch.	s-chan. γ
`WTSET(11)`	$\mathcal{O}(\alpha^0)$	Yes	Yes	Yes
`WTSET(12)`	$\mathcal{O}(\alpha^1)$	Yes	Yes	Yes
`WTSET(51)`	$\mathcal{O}(\alpha^0)$	No	No	No
`WTSET(52)`	$\mathcal{O}(\alpha^1)$	No	No	No
	special miscellaneous			
`WTSET(20)`	$\mathcal{O}(\alpha^1)$	Yes	No	No
`WTSET(21)`	`=WTSET(20)-WTSET(52)`			
`WTSET(22)`	$\mathcal{O}(\alpha^1)$	Yes	Yes	No
`WTSET(23)`	`=WTSET(22)-WTSET(20)`			
`WTSET(24)`	$\mathcal{O}(\alpha^1)$	Yes	Yes	Yes
`WTSET(25)`	`=WTSET(24)-WTSET(22)`			
`WTSET(26)`	$\tilde{\beta}_0$ component in `WTSET(20)`			
`WTSET(27)`	$\tilde{\beta}_1$ component in `WTSET(20)`,			
	i.e. `WTSET(20)=WTSET(26)+WTSET(27)`			
`WTSET(61)`	LL component in `WTSET(51)`			
`WTSET(62)`	Next-to-LL component in `WTSET(52)`,			

LUMLOG			
Entry	Type of non-singlet electron structure function		
	QED order	Exponentiation	light pairs
`WTSET(1)`	$\mathcal{O}(\alpha^0)$	Yes	No
`WTSET(2)`	$\mathcal{O}(\alpha^1)$	Yes	No
`WTSET(3)`	$\mathcal{O}(\alpha^2)$	Yes	No
`WTSET(4)`	$\mathcal{O}(\alpha^3)$	Yes	No
`WTSET(5)`	$\mathcal{O}(\alpha^3)$	Yes	Yes
`WTSET(11)`	$\mathcal{O}(\alpha^0)$	No	No
`WTSET(12)`	$\mathcal{O}(\alpha^1)$	No	No
`WTSET(13)`	$\mathcal{O}(\alpha^2)$	No	No

OLDBIS				
Entry	Corrections present in $\mathcal{O}(\alpha)$ matrix element			
	up-down interf.	Vac. pol.	Z-exch.	s-chan. γ
`WTSET(10)`	Yes	Yes	Yes	Yes
`WTSET(11)`	No	No	No	No
`WTSET(12)`	Yes	No	No	No
`WTSET(13)`	Yes	No	No	Yes

Table 4: *Explanation of parallel weights in* **WTSET** *list, separately for each sub-generator. The principal total weight* **WTMOD** *related to one of the entries depending on the input parameters. For BHLUM2 it corresponds to* **WTSET(2)** *or* **WTSET(52)** *depending on* **KEYPIA***, for LUMLOG it corresponds to* **WTSET(4)** *and for OLDBIS it corresponds to* **WTSET(10+KEYSIN)**.

Parameter	Meaning
NPAR(10)=NEVGEN	Number of generated Monte Carlo events
NPAR(20)=NEVGEN	Number of generated Monte Carlo events
XPAR(10)=XSEC	Total cross-section (nanobarns) resulting from the Monte Carlo integration
XPAR(11)=RXSEC	Statistical relative error of the cross-section in XPAR(10)
XPAR(20)	=XPAR(10) for KEYWGT=0, otherwise it is the crude Monte Carlo cross-section (nanobarns), to be used for renormalizing histograms
XPAR(21)	=XPAR(11) for KEYWGT=0, otherwise = 0 because crude cross-section is known exactly

Table 5: *List of output parameters in post-generation mode,* `MODE=1,2`.

more detail, separately for each type of sub-generator, in Table 4. It is important to know that for variable weight events the final state four-momenta may be ill-defined in events with `WTMOD=0` or `WTCRU1*WTCRU2=0`. In order to avoid an unnecessary crash of the user program the kinematical calculations should be protected by the appropriate `IF (WTMOD.NE.0D0) THEN ... ENDIF` conditional statement. This will also speed up a bit the user histogramming program.

4.3 Output in post-generation mode

After generating a series of Monte Carlo events the user may optionally call BHLUMI with `MODE=1` or `MODE=2` to obtain the value of the integrated cross-section (in nanobarn units) corresponding to the entire generated Monte Carlo sample through `XPAR(10)` and its relative statistical error from `XPAR(11)`. For constant weight events this cross-section will be used to calculate the cross-section after the trigger and to introduce absolute (in nanobarns) normalization of the histograms. For weighted events, the content of histograms has to be renormalized using the crude Monte Carlo cross-section (before Monte Carlo integration) and this quantity is available from `XPAR(20)`. We summarize the information on BHLUMI output parameters in post-generation mode in Table 5. Note that OLDBIS sub-generators fill in post-generation mode some additional entries of `XPAR` with quantities (soft and hard cross-section separately) which can be useful in the special tests of the type presented in Ref. [4]. We refer the reader to the corresponding Table in the source code of the OLDBIS for details.

4.4 Other recommendations for the user

Our program has the form of a typical Monte Carlo event generator. One Monte Carlo event is produced by the single call on the generator subprogram. The user is responsible for arranging the loop over the events and selecting/accepting events according to his favorite experimental trigger. Part of the demonstration program described in the next section, for instance subroutine `BOKER1`, may serve as a useful example for developing the user's own application program. The user may calculate the total cross-section in the standard way using the number of accepted events and the total cross-section provided by the event generator at the end of the Monte Carlo run. The Born cross-section for arbitrary experimental cuts can be calculated only with help of LUMLOG.

The primary trigger for the Monte Carlo sub-generator `BHLUM2` is defined by only two parameters, $t_{\rm min}$ and $t_{\rm max}$. Their values should be adjusted in such a way that the entire phase space defined by the secondary trigger of the user fits well inside the primary Monte Carlo trigger. It is shown in Fig. 4 of Ref. [2] how to do it in practice. The essential problem is to choose $t_{\rm min}$ not too high such that the entire phase space defined by the *experimental* trigger is covered, and not too low because it would lead to excessive rejection of Monte Carlo events. For the $\mathcal{O}(\alpha)$ and $\mathcal{O}(L^2\alpha^2)$ this procedure is completely safe. For more complicated sub-leading (second order) processes one may get a contribution from the $t = 0$ region (integrable, of the type $const \times \delta(t)$) which cannot be discovered by numerical exercises with variation of $t_{\rm min}$, as described in Ref. [2]. This kind of contribution is discussed in Ref. [36] and is found to be negligible. The other two generators generate events within user-defined angular range.

The change of random number seed is possible for RANMAR generator. The example is provided in the test BHLDE2.

The multiphoton sub-generator may become less efficient for $\vartheta_{\rm min} < 1^\circ$ because of many events with zero weight (mass weight). This may be cured by setting `TRMX2= CONST*TRMAX` in `BHLUM2`. The constant may be simply one but it should be adjusted such that the fraction of events with `WTMOD > WTMAX` is reasonably small. Practical requirement is that the cross-section of these events (printed by `BHLUM2`) should be kept at the reasonable level with respect precision aimed by the user. (Such adjustments can also make sense for angles higher then 1°.)

The following limitations of the BHLUMI 2.01 should be kept in mind:

- The program validity range is restricted to small scattering angles, $\vartheta < 10^\circ$.
- The $\gamma\gamma$ and 3γ events which enter into typical luminosity measurements should be discussed separately. Such contributions are small and can be calculated with help of the other Monte Carlo program [37]. The same remark concerns production of the four lepton final states.

5 Demonstration program and its output

The program deck includes two demonstration programs BHLDE1 and BHLDE2 followed by the Monte Carlo generator BHLUMI itself.

For a quick start we have prepared the simple demonstration program BHLDE1, see also the beginning of the previous section, which generates 10000 constant-weight multiphoton events. The printout from BHLDE1 is included in the beginning of the test-run-output. It starts with the printout of input parameters followed by the four-momenta of the first event and the final output from the multiphoton sub-generator BHLUM2 and from BHLDE1 itself. The final printout from BHLUM2 is divided into three windows. The first one includes the total number of the generated events, the total $\mathcal{O}(\alpha)$ exponentiated cross-section for all of the Monte Carlo sample with its statistical error. The constant-weight events are made (inside the generator) out of variable-weight events by the usual rejection using maximum weight `WTMAX`. In the *first window* the value of the `WTMAX` is printed together with the number of events with negative total weight (`WTMOD<0`) and the number of the "over-weighted" events (`WTMOD>WTMAX`). If "over-weighted" events are present then the crucial quantity to look at is not really the number of them but rather the cross-section from the part of the phase space where `WTMOD>WTMAX`. This quantity together with other statistics on weights is printed in the *second window*. The user may not worry too much about the *third* output window from BHLUM2 which shows averages of certain special test-weights controlling some important technical aspects of the Monte Carlo generation. All of them should be equal one within statistical error. Later in the test-run-output we see two histograms printed by BHLDE1.[25] The first one is the distribution of the $e^{\pm}$ scattering angle. The Monte Carlo multiphoton sub-generator generates events in the well-defined t-range $|t_{\rm min}| < |t| < |t_{\rm max}|$ of the t-channel transfer. Owing to bremsstrahlung this range does not correspond to a unique range in the scattering angle, as is clearly seen in the histogram. The second histogram shows the distribution of the total weight `WTMOD`.

For advanced users we have prepared the second demonstration program BHLDE2 which shows how to use all three sub-generators. It is derived directly from the program which was used to obtain all Monte Carlo results of Ref. [2]. It may be used for a wide variety of calculations and/or may serve as an example to be followed in the construction of the user's own application program. The demonstration program BHLDE2 is run three times for three different data sets. In each run it calls on a selected (by input flags) subset of the four testing programs `BOKER1` - `BOKER4` which perform tests specific to one or more sub-generators.

In the *first* test-run BHLDE2 generates 20000 variable-weight events using multiphoton sub-generator BHLUM2. In the test-run-output we see, first, printout of all input parameters, next, four-momenta of the first two events and, finally, the post-generation output from BHLUM2 and from two testing subprograms `BOKER1` and `BOKER4`. The cross-section from BHLUM2 corresponds to the entire Monte Carlo sample while `BOKER1,4` calculate and print the cross-sections and corrections for events accepted by the trigger. In all BHLDE2 tests the trigger is of the calorimetric type, i.e. it does not distinguish electrons and photons. It is embodied in routine `TRIGAS1` and it is precisely the trigger of Ref. [2]. All Monte Carlo calculations in

[25]The program contains much more histogramming but the rest of it is commented-out. The reader is reminded that in the Monte Carlo runs with high statistics (10~7 events) Monte Carlo the histogramming package has to use double precision arithmetic in storing histogram content, Ref. [4].

this test-run correspond closely to those of Ref. [2].

In the *second* test-run LUMLOG generates 40000 variable-weight events using leading-logarithmic sub-generator LUMLOG. In the test-run-output we have, first, printout of all input parameters followed by four-momenta of the first two events and, next, the post-generation output from LUMLOG and from one testing subprogram `BOKER3`. The cross-section printed by LUMLOG corresponds to the entire Monte Carlo sample while `BOKER1` calculates and prints results for events accepted by the trigger. The calculations in this test run correspond closely to those of Ref. [3], they also provide some results for inter-generator tests of Ref. [2].

In the *third* test-run OLDBIS generates 50000 variable-weight events using $\mathcal{O}(\alpha)$ sub-generator OLDBIS, which is a modified/improved version of the OLDBAB Monte Carlo of Ref. [5], see also Ref. [4]. In the test-run-output we find, first, printout of all input parameters followed by four-momenta of the first two events and, next, the post-generation output from OLDBIS and from two testing subprograms `BOKER1` and `BOKER2`. The cross-section printed by OLDBIS corresponds to the entire Monte Carlo sample while `BOKER1,2` calculate and print results for events accepted by the trigger. Calculations in this test run give the value of the up-down interference contributions with and without the experimental trigger, see Tables in Ref. [4] and Ref. [2]. This test may also serve to check[26] if this version of OLDBIS reproduces properly the true $\mathcal{O}(\alpha)$ QED cross-section presented in Table 1 of Ref. [4]. The agreement within 0.02% technical precision is expected.

6 Conclusions

The program BHLUMI 2.01 represents the state of the art in the calculations of the QED corrections to small-angle Bhabha and their technical and physical precision. On the one hand, it is a full-scale stand-alone Monte Carlo event generator ready to use for any kind of detector study and to calculate the QED correction to luminosity measurement. On the other hand, it is a tool box of programs which allows one to do all kind of sophisticated tests and to cross-check in many ways the obtained numerical results. It is not the final ideal solution of the problem and a lot of things in the program could and should be improved. The present program provides very solid starting point for new improvements and developments.

Acknowledgments

Useful discussions with Drs. S. Arcelli, M. Dallavalle, B. Pietrzyk, W.D. Schlatter and M. Skrzypek are acknowledged. Two of us (BFLW and SJ) acknowledge the support and kind hospitality of Dr. J. Ellis of the CERN Theory Division and three of us (SJ, ERW and ZW) are grateful to the ALEPH and OPAL Collaboration for support.

[26]For such a check the user has to adjust values of $\vartheta_{\rm min,max}$ and set `KEYTRI=0` in the input data – this will replace `TRIGAS1` with the simpler trigger `TRIGAS0` of Ref. [4].

References

[1] Z. Wąs and S. Jadach, "QED predictions and their systematic errors", MC91-Workshop on Detector and Event Simulation in High Energy Physics, Amsterdam 1991, page 438, also CERN preprint TH-6159 (1991).

[2] S. Jadach, E. Richter-Wąs, B.F.L. Ward and Z. Wąs, Phys. Lett. **B** (1991) in press (CERN preprint TH-6118).

[3] S. Jadach, E. Richter-Wąs, B.F.L. Ward and Z. Wąs, Phys. Lett. **B260** (1991) 438 (CERN preprint TH-5995).

[4] S. Jadach, E. Richter-Wąs, B.F.L. Ward and Z. Wąs, Phys. Lett. **B253** (1991) 469 (CERN preprint TH-5888).

[5] F.A. Berends and R. Kleiss, Nucl. Phys. **B228** (1983) 537.

[6] F.A. Berends, K.J.F. Gaemers and R. Gastmans, Nucl. Phys. **B57** (1973) 381; Nucl. Phys. **B68** (1974) 541.

[7] F.A. Berends and G.J. Komen, Nucl. Phys. **B115** (1976) 114.

[8] M. Böhm, A. Denner and W. Hollik, Nucl. Phys. **B304** (1988) 687; F.A. Berends, R. Kleiss and W. Hollik, Nucl. Phys. **B304** (1988) 712.

[9] S. Jadach and B. F. L. Ward, Phys. Rev. **D40** (1989) 3582.

[10] S. Jadach, M. Skrzypek and B.F.L. Ward, Phys. Lett. **B257** (1991) 173.

[11] F.A Berends, W.L. Van Neerven and G.J.H. Burgers, Nucl. Phys. **B297** (1988) 429.

[12] D.R. Yennie, S. Frautschi and H. Suura, Ann. Phys. (NY) **13** (1961) 379.

[13] S. Jadach, E. Richter-Wąs, B.F.L. Ward and Z. Wąs, Phys. Rev. **D44** (1991) 2669.

[14] S. Jadach and B.F.L. Ward, "Exclusive exponentiation in the Monte Carlo Yennie-Frautschi-Suura approach", in Proc. Sussex University Conference "Electroweak Physics", eds. N. Dombey and F. Boudjema (Plenum Publ. Co., London, 1989).

[15] J.D. Jackson and D.L. Scharre, Nucl. Instrum. Meth. **128**, 13 (1975).

[16] E.A. Kuraev and V.S. Fadin, Sov. J. Nucl. Phys. **41** (1985) 466.

[17] S. Jadach, M. Skrzypek and B.F.L. Ward, Phys. Lett. **B257** (1991) 173.

[18] S. Jadach, "Yennie-Frautschi-Suura soft photons in Monte Carlo event generators",
preprint of MPI-München, MPI-PAE/PTh 6/87, Jan. 1987.

[19] S. Jadach and B.F.L. Ward, Phys. Rev. **D38** (1988) 2897.

[20] S. Jadach and B.F.L. Ward, Comput. Phys. Commun. **56** (1990) 351.

[21] S. Jadach, Z. Wąsand B.F.L. Ward, Comput. Phys. Commun. **66** (1991) 276.

[22] S. Jadach et.al., "YFS3 Monte Carlo for LEP/SLC with second order initial and final state bremsstrahlung", S. Jadach et.al. (1990), unpublished.

[23] R. Kleiss, "Monte Carlo exponentiation and final-state radiative corrections", preprint CERN-TH.6155, (1991).

[24] S. Jadach and W. Płaczek, "The Monte Carlo program LESKO-F for deep inelastic $e^{\pm} \rightarrow e^{\pm}X$ scattering at HERA including QED bremsstrahlung from lepton line", DESY-91-031 and, TPJU-4-91 preprint (1991), Comput. Phys. Commun. in print.

[25] V. Sudakov, Zh. Eksp. Teor. Fiz. **30** (1956) 187; Sov. Phys. JETP **3** (1956) 115.

[26] W. Beenakker, F. A. Berends and S. C. van der Marck, Nucl. Phys. **B355** (1991) 281.

[27] S. Jadach, Acta Phys. Polon., **B16** (1985) 1007.

[28] F.A. Berends, R. Kleiss and S. Jadach, Nucl. Phys., **B202** (1982) 63; Comput. Phys. Commun. **29** (1983) 185.

[29] V.N. Gribov and L.N. Lipatov, Yad. Fiz. **15** (1972) 781; Sov. J. Nucl. Phys. **15** (1972) 438.

[30] M. Skrzypek and S. Jadach, Z. Phys. **C49** (1991) 577.

[31] W. Beenakker, F.A. Berends and S.C. van der Marck, Nucl. Phys. **B355** (1991) 281.

[32] S. Jadach, M. Skrzypek and B.F.L. Ward, "Exponentiation, higher orders and leading logs", Jagellonian Univ. preprint TPJU-02/90 (1990), Proc. Rencontre de Moriond, Les Arcs, 1990.

[33] H. Burkhardt et. al., Z. Phys. **C43** (1989) 497.

[34] F. James, Comput. Phys. Commun. **60** (1990) 329; G. Marsaglia, A. Zaman and W.-W. Tsang, Stat. Prob. Lett. **9** (1990) 35.

[35] P. L'Ecuyer, Commun. ACM **31** (1988) 742.

[36] R. Kleiss, "Magnitude of luminosity cross-section with lost fermions at LEP", preprint CERN-TH.6141 (1991).

[37] F.A. Berends and R. Kleiss, Nucl. Phys. **B186** (1981) 22.

TEST RUN OUTPUT

```
=============================================
==========********************==============
==========***  B H L D E 1  ***=============
==========********************==============
=============================================
========================================================================
=                                                                      =
=              BBB   B  B  B    B   B  B     B  B                      =
=              B  B  B  B  B    B   B  BB   BB  B                      =
=              BBB   BBBB  B    B   B  B BB B   B                      =
=              B  B  B  B  B    B   B  B     B  B                      =
=              BBB   B  B  BBBB  BBB   B     B  B                      =
=                                                                      =
=              ********************************                        =
=              *   BHLUMI version 2.01        *                        =
=              *   September      1991        *                        =
=              *         AUTHORS              *                        =
=              * S. Jadach, E. Richter-Was    *                        =
=              *    B.F.L. Ward, Z. Was       *                        =
=              ********************************                        =
========================================================================
========================================================================
=               This program is based on papers                        =
=               --------------------------------                       =
=               Long-write-up, sept.91 TH-6230                         =
=               P.L. (1991) in print,  TH-6118.                        =
=               Phys. Rev. D40 (1989) 3582.                            =
=               P.L. B260 (1991) 173,  TH-5995.                        =
=               P.L. B253 (1991) 469,  TH-5888.                        =
=               Nucl. Phys. B228 (1983) 537.                           =
========================================================================
========================================================================
=              ********************************                        =
=                BHLUM2: INPUT PARAMETRES                              =
=              ********************************                        =
=           3001                 options   switch       KEYOPT     N1 =
=              1                 VARRAN    switch       KEYRND        =
=              0                 weighting switch       KEYWGT        =
=              1                 radiation switch       KEYRAD     N2 =
=              1                 vac_pol   switch       KEYPIA        =
=    92.00000000                 CMS enegry  [GeV]      CMSENE     X1 =
=   8464.0000                    CMSENE^2  [GeV^2]      SVAR          =
=   1.9000000                    trasf_min [GeV^2]      TRMIN      X2 =
=   64.000000                    trasf_min [GeV^2]      TRMAX      X3 =
=  0.22448015E-03                xi_min=TRMIN/SVAR      XIMIN         =
=  0.75614367E-02                xi_max=TRMAX/SVAR      XIMAX         =
=    29.96644459                 theta_min  [mrad]      THMIN         =
=   174.13296469                 theta_max  [mrad]      THMAX         =
=     1.71695080                 theta_min   [deg]      THMIN         =
=     9.97708395                 theta_max   [deg]      THMAX         =
=     0.00010000                 eps_CM infr. cut       EPSCM      X4 =
=     0.00000002                 delta  infr. cut       DEL           =
=    -0.01882600                 RePi(transf_min)                     =
=    -0.03447518                 RePi(transf_max)                     =
========================================================================
===================DUMPS===================
 P2    0.2741897459098   2.1723085569035  45.9477038139964  45.9998433749847
 Q2   -0.2741913701105  -2.1722864183026 -45.9474506578303  45.9995894699676
SUM   -0.0000016242007   0.0000221386009   0.0002531561661  91.9994328449523
========================================================================
=              ********************************                        =
=                BHLUM2:        WINDOW A                               =
=              ********************************                        =
=          10000                 Accepted total         NEVGEN     A1 =
=          27167                 Raw prior reject.      IEVENT     A2 =
=    136.99717      +- 0.84479971 Xsec M.C. [nb]        XSECMC     A3 =
=     0.00616655                 relat. error           ERELMC     A4 =
=     1.02951834    +- 0.00616655 weight  M.C.          AWT        A5 =
=              0                 WT<0                   NEVNEG     A6 =
=              0                 WT>WTMAX               NEVOVE     A7 =
=     2.80000000                 Maximum WT             WWMX       A8 =
========================================================================
========================================================================
=              ********************************                        =
=                BHLUM2:        WINDOW B                               =
=              ********************************                        =
=     0.25202302  +- 0.00583426  WT1*WT2*T/TP*T/TQ                 B1 =
=     0.99981202  +- 0.00008408  WT3 from KINO4                     B2 =
=     4.25421154  +- 0.00088845  YFS formfac            WT4         B4 =
=     1.02951834  +- 0.00616655  TOTAL                              B5 =
=     0.03972234  +- 0.00038967  WT-WTLL                WT5         B6 =
=     0.00000000  +- 0.00000000  xsec/xtot: WT>WTMAX    WT5         B7 =
========================================================================
========================================================================
=              ********************************                        =
=                          WINDOW C                                    =
=              Built-in average control weights.                       =
=              Should equal one +- statist. err.                       =
=              ********************************                        =
=     0.99597895  +- 0.00771405  <WCTA1>                            C1 =
=     1.00881455  +- 0.00763359  <WCTA2>                            C2 =
=     0.99724723  +- 0.01455639  <WCTA1*WCTA2>                      C3 =
=     0.99729506  +- 0.00364966  <WCTB1>                            C4 =
=     1.00078773  +- 0.00362024  <WCTB2>                            C5 =
=     0.99883120  +- 0.00557219  <WCTB1*WCTB2>                      C6 =
========================================================================
```

```
THETA DISTRIBUTION (DEGREES)
HBOOK     ID =       100                                        DATE  28/09/91
    3500                -
    3400                I
    3300                I
    3200                I
    3100                I
    3000                I
    2900                I
    2800                I
    2700                I
    2600                I-
    2500                II
    2400                II
    2300                II
    2200                II
    2100                II
    2000                II-
    1900                I I
    1800                I I
    1700                I I
    1600               -I I-
    1500               I   I
    1400               I   I
    1300               I   I-
    1200               I    I
    1100               I    I
    1000               I    I-
     900               I     I-
     800               I      I-
     700               I       I
     600               I       I-
     500               I        I--
     400               I          I-
     300               I           I----
     200               I               I-------
     100           - --I                      I----------------------------
CHANNELS  10   0         1         2         3         4         5         6
           1   123456789012345678901234567890123456789012345678901234567890
CONTENTS1000           132111
         100           545952987544322221111111
          10           841716224897786349854220877666744665332333
          1.      1 6866164568492356763775300658456218842245682685666662543
LOW-EDGE  10                                                     1111111111
          1.        1111122222333334444455555666667777788888999990000011111
           0   024680246802468024680246802468024680246802468024680246802468
* ENTRIES =       20000      * ALL CHANNELS = 0.1983E+05   * UNDERFLOW = 0.0000E+00
                                                          * OVERFLOW = 0.1730E+03
* BIN WID = 0.2000E+00       * MEAN VALUE   = 0.3027E+01   * R . M . S = 0.1573E+01
BHLUM2, WEIGHT DISTRIBUTION
HBOOK     ID =      9001                                        DATE  28/09/91
    1240            -
    1200            I
    1160            I
    1120            I
    1080            I
    1040            I
    1000            I
     960            I
     920            I
     880            I
     840            I
     800            I
     760            I
     720            I
     680            I
     640            I
     600            I
     560            I
     520            I
     480            I
     440            I
     400            I           -
     360            I           I
     320            I          -I
     280            I          II
     240            I          II
     200            I         -II
     160            I         I I
     120            I      ---I I
      80            I------I    I
      40            I           I-
CHANNELS  10   0         1         2         3         4
           1   1234567890123456789012345678901234567890
CONTENTS1000        1
*10**- 1 100        2        1133
          10        2744566881807
          1.        3388257451023
           0   0000022020674594592000000000000000000000
LOW-EDGE       ------
          1.   1         111112222233333444445555566666
           0   0864202468024680246802468024680246802468
* ENTRIES =       27167      * ALL CHANNELS = 0.2717E+05   * UNDERFLOW = 0.0000E+00
                                                          * OVERFLOW = 0.0000E+00
* BIN WID = 0.2000E+00       * MEAN VALUE   = 0.1069E+01   * R . M . S = 0.1008E+01
```

```
=============================================
=========*****************************=======
==========***  B H L D E 2  ***==============
=========*****************************=======
=============================================
=============== Data set for BHLUM2 test ====================
 KAT1  KAT2  KAT3  KAT4  KAT5  KAT6
    1     0     0     1     0     0
      NEVT      KEYRAD      KEYOPT      KEYTRI
     20000           0        3011           1
    CMSENE       TMING       RAXIG      VMAXG         XKO
 92.500000    2.700000    6.714420   0.999900    0.000100
     TMINW       RAXIW       TMINN      RAXIN       VMAXE
  2.700000    6.714420    3.300000   3.641960    0.500000
     TMING       TMAXG       TMINW      TMAXW       TMINN       TMAXN
  2.700000    6.999999    2.700000   6.999999    3.300000    6.299996
===========================================================================
=               *********************************                         =
=                 BHLUM2: INPUT PARAMETRES                                =
=               *********************************                         =
=             3011                options   switch        KEYOPT     N1 =
=                1                VARRAN    switch        KEYRND          =
=                1                weighting switch        KEYWGT          =
=                0                radiation switch        KEYRAD     N2 =
=                0                vac_pol   switch        KEYPIA          =
=       92.50000000               CMS enegry  [GeV]       CMSENE     X1 =
=     8556.2500                   CMSENE^2  [GeV^2]       SVAR            =
=     1.8997015                   trasf_min [GeV^2]       TRMIN      X2 =
=     63.776968                   trasf_min [GeV^2]       TRMAX      X3 =
=    0.22202501E-03               xi_min=TRMIN/SVAR       XIMIN           =
=    0.74538458E-02               xi_max=TRMAX/SVAR       XIMAX           =
=       29.80211008               theta_min  [mrad]       THMIN           =
=      172.88654864               theta_max  [mrad]       THMAX           =
=        1.70753513               theta_min   [deg]       THMIN           =
=        9.90566957               theta_max   [deg]       THMAX           =
=        0.00010000               eps_CM infr. cut        EPSCM      X4 =
=        0.00000002               delta  infr. cut        DEL             =
=       -0.01882542               RePi(transf_min)                        =
=       -0.03445734               RePi(transf_max)                        =
===========================================================================
===========================================================================
=               BOKER1 start initialisation....                           =
=        0.04712389              theta_min wide           TH1W            =
=        0.12217303              theta_max wide           TH2W            =
=        0.05759587              theta_min narrow         TH1N            =
=        0.10995568              theta_max narrow         TH2N            =
=                 6              No of phi-sectors        NPHI            =
=       46.63570105              Born Wide    [nb]        BORWID          =
=                 TH.6118 Tab.2a line.1                                   =
=       26.60621117              Born Narrow  [nb]        BORNAR          =
=                 .... BOKER1 end initialization                          =
===========================================================================
===========================================================================
=               BOKER4 start initialisation....                           =
=               *********************************                         =
=               BHLUM2 related tests...                                   =
=               total the best correction.                                =
=               test of phase space completness                           =
=               transfer dist. before/after trig.                         =
=               *********************************                         =
=        0.04712389              theta_min wide           TH1W            =
=        0.12217303              theta_max wide           TH2W            =
=        0.05759587              theta_min narrow         TH1N            =
=        0.10995568              theta_max narrow         TH2N            =
=                 6              No of phi-sectors        NPHI            =
=       46.63570105              Born Wide    [nb]        BORWID          =
=       26.60621117              Born Narrow  [nb]        BORNAR          =
=        4.74925368              trans_min Born           TRMINB          =
=        2.37462684              trans_min L-Log          TRMINL          =
=       31.88848388              trans_max L-Log          TRMAXL          =
=       -0.02234805              RePi(tran_min_Born)      REPIB           =
=       -0.01966016              RePi(tran_LL_min)        REPI1           =
=       -0.03096322              RePi(tran_LL_max)        REPI2           =
=                 .... BOKER4 end initialization                          =
===========================================================================
====================DUMPS====================
 P2   -2.1239698336798  -1.1511191390799  30.2356225518484  30.3319830292351
 Q2    3.2115846561887   1.7244408529133 -46.0927473976456  46.2366665665190
PHO   -1.0878320349693  -0.5737416002160  15.8600829110339  15.9076958703954
PHO    0.0002177657336   0.0004192392096  -0.0131973408617   0.0132057937758
PHO   -0.0000001586912  -0.0000000414065   0.0103308522715   0.0103308522728
SUM    0.0000003945821  -0.0000006885795   0.0000915766466  92.4998821121981
====================DUMPS====================
 P2    0.6308119706479   4.6016349927299   0.3321862314836   4.6565347955576
 Q2   -1.9046336117442  -7.1969718912237 -43.6172676607033  44.2480516186273
PHO    1.2398073958553   2.3515903641589  43.2670502963703  43.3486417430611
PHO    0.0339990478108   0.2436361554497   0.0180261763712   0.2466565517657
SUM   -0.0000151974302  -0.0001103788852  -0.0000049564782  92.4998847090117
```

```
===========================================================================
=              *********************************                          =
=                BHLUM2:        WINDOW A                                  =
=              *********************************                          =
=          20000                 Accepted total         NEVGEN      A1 =
=          20000                 Raw prior reject.      IEVENT      A2 =
=     130.73587     +- 0.94512918   Xsec M.C. [nb]      XSECMC      A3 =
=       0.00722930               relat. error           ERELMC      A4 =
=       0.98241140  +- 0.00722930   weight  M.C.        AWT         A5 =
=              0                 WT<0                   NEVNEG      A6 =
=              0                 WT>WTMAX               NEVOVE      A7 =
=       2.80000000               Maximum WT             WWMX        A8 =
===========================================================================
===========================================================================
=              ============ BOKER1 ============                           =
=              *********************************                          =
=              Integrated x-sect. various cuts                            =
=                    Any sub-generator                                    =
=                    PURE BREMSSTRAHLUNG                                  =
=              th.6118 fig.2a, middle bin                                 =
=              th.6118 fig.3,  middle bin                                 =
=              *********************************                          =
=                     ////// wide /////                                   =
=      44.36837361  +- 0.68942780  xsec. total          XS                =
=       0.01553872                 rel. error           DS/XS             =
=      -0.04861785  +- 0.01478326  O(alfi)/Born -1      RXS               =
=                    ///// narrow /////                                   =
=      25.07477798  +- 0.53708095  xsec. total          XS                =
=       0.02141917                 rel. error           DS/XS             =
=      -0.05755924  +- 0.02018630  O(alfi)/Born -1      RXS               =
=                     ////// mixed /////                                  =
=              th.6118 Tab.2a line.2, for BHLUM2                          =
=              th.6118 Tab.2b line.2 (!!!)                                =
=      26.37487637  +- 0.38625084  xsec. total          XS                =
=       0.01464465                 rel. error           DS/XS             =
=      -0.00869477  +- 0.01451732  O(alfi)/Born -1      RXS               =
===========================================================================
===========================================================================
=              ============ BOKER4 ============                           =
=                    BHLUM2 related tests                                 =
=              Various corr. Table 2 in TH.6118                           =
=              *********************************                          =
=              Integrated x-sect. total best                              =
=              th.6118 fig.1, middle bin                                  =
=              *********************************                          =
=                     ////// wide /////                                   =
=      46.27248559  +- 0.71901877  xsec. total          XS                =
=       0.01553880                 rel. error           DS/XS             =
=      -0.00778836  +- 0.01541778  O(alfi)/Born -1      RXS               =
=                    ///// narrow /////                                   =
=      26.16234991  +- 0.56037609  xsec. total          XS                =
=       0.02141918                 rel. error           DS/XS             =
=      -0.01668262  +- 0.02106185  O(alfi)/Born -1      RXS               =
=                     ////// mixed /////                                  =
=              th.6118 tab.2a line.6                                      =
=      27.51799030  +- 0.40299416  xsec. total          XS                =
=       0.01464475                 rel. error           DS/XS             =
=       0.03426941  +- 0.01514662  O(alfi)/Born -1      RXS               =
=              *********************************                          =
=              Integrated x-sect. SUBLEADING                              =
=              th.6118 fig.3, middle bin                                  =
=              *********************************                          =
=                     ////// wide /////                                   =
*      -0.00183021  +--0.00044311  O(alfi-alfj)/Born    RXS               *
=                    ///// narrow /////                                   =
*      -0.00201953  +--0.00056244  O(alfi-alfj)/Born    RXS               *
=                     ////// mixed /////                                  =
*      -0.00215292  +--0.00044849  O(alfi-alfj)/Born    RXS               *
=              *********************************                          =
=              Integrated x-sect. Vac_Pol                                 =
=              *********************************                          =
=                     ////// wide /////                                   =
*       0.04965675  +- 0.00077260  O(alfi-alfj)/Born    RXS               *
=                    ///// narrow /////                                   =
*       0.05128055  +- 0.00109973  O(alfi-alfj)/Born    RXS               *
=                     ////// mixed /////                                  =
*       0.05373504  +- 0.00078871  O(alfi-alfj)/Born    RXS               *
=                     ////// mixed /////                                  =
=              th.6118 tab.2a line.3                                      =
=      27.80456211  +- 0.40718648  xsec. total          XS                =
=       0.01464459                 rel. error           DS/XS             =
=       0.04504027  +- 0.01530419  O(alfi)/Born -1      RXS               =
```

```
=              *********************************                     =
=              Integrated x-sect. Z contr.                           =
=              *********************************                     =
=                     ////// wide /////                              =
*     -0.00770050  +--0.00013817  O(alfi-alfj)/Born        RXS       *
=                     ///// narrow /////                             =
*     -0.00907494  +--0.00020783  O(alfi-alfj)/Born        RXS       *
=                     ////// mixed /////                             =
*     -0.00939510  +--0.00014838  O(alfi-alfj)/Born        RXS       *
=                     ////// mixed /////                             =
=              th.6118 tab.2a line.4                                 =
=     27.55459398  +- 0.40352921  xsec. total              XS        =
=      0.01464472                 rel. error               DS/XS     =
=      0.03564517  +- 0.01516673  O(alfi)/Born -1          RXS       =
=              *********************************                     =
=              Integrated x-sect. s_Cor.                             =
=              *********************************                     =
=                     ////// mixed /////                             =
=              th.6118 tab.2a line.5                                 =
*     -0.00137576  +--0.00002176  O(alfi-alfj)/Born        RXS       *
=              *********************************                     =
=              Beta_0, Beta_1 contr.                                 =
=              th.6118 tab.2b indirectly                             =
=              *********************************                     =
=                     ////// mixed /////                             =
*      1.03273335  +- 0.01514252  O(alfi-alfj)/Born        RXS       *
*      0.00153606  +- 0.00015224  O(alfi-alfj)/Born        RXS       *
=              *********************************                     =
=              WT>WTMAX contr. important x-check                     =
=              *********************************                     =
=                     ////// mixed /////                             =
*      0.00000000  +- 0.00000000  O(alfi-alfj)/Born        RXS       *
======================================================================
==============================================
==========**********************==============
==========***  B H L D E 2  ***===============
==========**********************==============
==============================================
==================== Data set for BHLOG test ================
  KAT1  KAT2  KAT3  KAT4  KAT5  KAT6
     0     0     1     0     0     0
        NEVT      KEYRAD      KEYOPT      KEYTRI
       40000         103        2011           1
      CMSENE       TMING       RAXIG       VMAXG         XKO
   92.500000    2.700000    6.714420    0.999900    0.000100
       TMINW       RAXIW       TMINN       RAXIN       VMAXE
    2.700000    6.714420    3.300000    3.641960    0.500000
       TMING       TMAXG       TMINW       TMAXW       TMINN       TMAXN
    2.700000    6.999999    2.700000    6.999999    3.300000    6.299996
**********************************************************************
*              *********************************                     *
*              *     *****************         *                     *
*              *     ***   LUMLOG  ***         *                     *
*              *     *****************         *                     *
*              *(P.L. B260 (1991) 173, TH-5995)*                     *
*              * This program is now part of   *                     *
*              *   BHLUMI version 2.01         *                     *
*              *   September      1991         *                     *
*              *         AUTHORS               *                     *
*              * S. Jadach, E. Richter-Was     *                     *
*              *    B.F.L. Ward, Z. Was        *                     *
*              *********************************                     *
**********************************************************************
**********************************************************************
*                 ===== LUMLOG/BHALOG  ======                        *
*                 initialization starts....                          *
*           2011                options   switch       KEYOPT     N1 *
*              1                VARRAN    switch       KEYRND        *
*              1                NOT IMPLEMENTED        KEYWGT        *
*            103                radiation switch       KEYRAD     N2 *
*              0                KEYTES                 KEYTES        *
*              3                KEYBLO                 KEYBLO        *
*              1                KEYRND                 KEYRND        *
*     92.50000000               CMSENE [GeV]           CMSENE     X1 *
*      2.70000000               TMIN  [degr.]          TMIN       X2 *
*      6.99999878               TMAX  [degr.]          TMAX       X3 *
*      0.00010000               xk0 cut (no exp.)      XKO        X4 *
*      0.99990000               minimum sprim/s        XKMAX      X5 *
*      6.71442000               RAXI                   RAXI          *
*      0.00055506               XIA=(1-cosTMINL)/2     XIA           *
*      0.00372692               XIB=(1-cosTMAXL)/2     XIB           *
*      0.03882886               ETA                                  *
*      0.00010000               ZMIN                                 *
*     46.69364390               Born crude             BORNC         *
*     46.63570105               Born exact             BORNX         *
*               end of initialization                                *
**********************************************************************
=====================DUMPS====================
 P2    -2.7174630565165   0.0707661030094  46.1700431740661  46.2500000000000
 Q2     2.7174630565165  -0.0707661030094 -46.1700431740661  46.2500000000000
SUM     0.0000000000000   0.0000000000000   0.0000000000000  92.5000000000000
=====================DUMPS====================
 P2    -1.1129412421130  -2.7717281451051  46.1534547448101  46.2500000000000
 Q2     1.1129412421130   2.7717281451051 -46.1534547448101  46.2500000000000
SUM     0.0000000000000   0.0000000000000   0.0000000000000  92.5000000000000
```

```
***************************************************************************
*                *********************************                        *
*                        OUTPUT FROM                                      *
*                  LUMLOG/BHALOG: WINDOW A                                *
*                *********************************                        *
*                   X.sect. in [nb] units                                 *
*                   for total generated sample                            *
*         40000                generated events         NEVGEN     A1 *
*      0.93251191  +- 0.00091453  vesko2 int. estim.       AWT        A2 *
*     45.72094639                 crude xsec (vesko2)      XCRU       A3 *
*     46.63570105                 Born xsection            BORN       A4 *
*                        ---  O(alf0)exp ---                              *
*     46.16985567  +- 0.06474979  xsec. total              XS01       A5 *
*      0.00140243                 rel. error                              *
*     -0.00998903                 O(alf0)/Born-1                          *
*                        ---  O(alf1)exp ---                              *
*     44.63913603  +- 0.06103672  xsec. total              XS02       A6 *
*      0.00136734                 rel. error                              *
*     -0.04281194                 O(alf1)/Born-1                          *
*     -0.03282291  +--0.00055951  O(alf1-alf0)/Born        RXS12          *
*                        ---  O(alf2)exp ---                              *
*     44.65897520  +- 0.06102473  xsec. total              XS03       A7 *
*      0.00136646                 rel. error                              *
*     -0.04238654                 O(alf2)/Born-1                          *
*      0.00042541  +- 0.00000535  O(alf2-alf1)/Born        RXS13          *
*                        ---  O(alf3)exp ---                              *
*     44.65928852  +- 0.06102421  xsec. total              XS04       A8 *
*      0.00136644                 rel. error                              *
*     -0.04237982                 O(alf3)/Born-1                          *
*      0.00000672  +- 0.00000016  O(alf3-alf2)/Born        RXS14          *
*                        ---  O(alf3)exp ---                              *
*                        ---  plus pairs ---                              *
*     44.64840847  +- 0.06073664  xsec. total              XS05       A9 *
*      0.00136033                 rel. error                              *
*     -0.04261312                 O(alf3+prs)/Born-1                      *
*     -0.00023330  +--0.00004603  pairs/Born               RXS15          *
***************************************************************************
===========================================================================
=                ============ BOKER3 =============                        =
=                LUMLOG corrections and  x-sects.                         =
=                *********************************                        =
=                         BORN,     O(alf0)                               =
=                *********************************                        =
=                      ////// wide /////                                  =
=     46.16924302  +- 0.23341144  xsec. total              XS             =
=      0.00505556                 rel. error               DS/XS          =
=     -0.01000217  +- 0.00500499  O(alfi)/Born -1          RXS            =
=                     ///// narrow /////                                  =
=     26.23556454  +- 0.20980608  xsec. total              XS             =
=      0.00799701                 rel. error               DS/XS          =
=     -0.01393083  +- 0.00788561  O(alfi)/Born -1          RXS            =
=                *********************************                        =
=                     O(alf3) exp. various cuts                           =
=                th.5995 fig.2, middle bin                                =
=                 th.5995 tab.2, third column                             =
=                *********************************                        =
=                      ////// wide /////                                  =
=     44.44232140  +- 0.06330455  xsec. total              XS             =
=      0.00142442                 rel. error               DS/XS          =
=     -0.04703220  +- 0.00135743  O(alfi)/Born -1          RXS            =
=                     ///// narrow /////                                  =
=     24.78129027  +- 0.11993425  xsec. total              XS             =
=      0.00483971                 rel. error               DS/XS          =
=     -0.06859003  +- 0.00450775  O(alfi)/Born -1          RXS            =
=                      ////// mixed /////                                 =
=     25.90836637  +- 0.08422933  xsec. total              XS             =
=      0.00325105                 rel. error               DS/XS          =
=     -0.02622864  +- 0.00316578  O(alfi)/Born -1          RXS            =
=                *********************************                        =
=                      O(alf3)-O(alf1) exp.                               =
=                th.5995 fig.4, middle bin                                =
=                *********************************                        =
=                      ---- WIDE -----                                    =
*      0.00039659  +- 0.00000503  O(alfi-alfj)/Born        RXS            *
=                      ---- NARROW ---                                    =
*      0.00033728  +- 0.00000621  O(alfi-alfj)/Born        RXS            *
=                      ---- MIXED  ---                                    =
*      0.00047084  +- 0.00000533  O(alfi-alfj)/Born        RXS            *
=                *********************************                        =
=                  O(alf3)exp-O(alf1)not.exp.                             =
=                th.5995 tab.2,  forth column                             =
=                th.5995 fig.3,  middle bin                               =
=                th.6118 fig.2a, middle bin                               =
=                *********************************                        =
=                      ---- WIDE -----                                    =
*      0.00048288  +- 0.00264221  O(alfi-alfj)/Born        RXS            *
=                      ---- NARROW ---                                    =
*      0.00201097  +- 0.00353207  O(alfi-alfj)/Born        RXS            *
=                      ---- MIXED  ---                                    =
=                th.6118 tab.2b line.2 (component)                        =
*     -0.00340174  +--0.00256015  O(alfi-alfj)/Born        RXS            *
```

```
=              *********************************                   =
=                O(alf3)exp-O(alf2)not.exp.                        =
=              *********************************                   =
=                     ---- MIXED  ---                              =
=              th.6118 tab.2b line.4                               =
*       0.00110501  +- 0.00082605  O(alfi-alfj)/Born       RXS     *
=              *********************************                   =
=                O(alf3)pairs-O(alf3)nopairs                       =
=              *********************************                   =
=                     ---- MIXED  ---                              =
=              th.6118 tab.2b line.6                               =
*      -0.00018526  +--0.00004366  O(alfi-alfj)/Born       RXS     *
====================================================================
==========================================
==========*****************************===
==========***  B H L D E 2  ***===========
==========*****************************===
==========================================
============= Data deck for BHLBIS test ======================
  KAT1  KAT2  KAT3  KAT4  KAT5  KAT6
     1     1     0     0     0     0
        NEVT      KEYRAD      KEYOPT      KEYTRI
       50000           1        1011           1
      CMSENE       TMING       RAXIG       VMAXG         XK0
   92.500000    1.700000  100.000000    0.999900    0.000100
       TMINW       RAXIW       TMINN       RAXIN       VMAXE
    2.700000    6.714420    3.300000    3.641960    0.500000
       TMING       TMAXG       TMINW       TMAXW       TMINN       TMAXN
    1.700000   17.062353    2.700000    6.999999    3.300000    6.299996
********************************************************************
*                   **********************                         *
*                   **    O L D B I S   **                         *
*                   **********************                         *
*              O(alpha) Monte Carlo Program                        *
*              for the small-angle Bhabha scat.                    *
*                      --------------                              *
*                      Important Note                              *
*                      --------------                              *
*              This is an improved version                         *
*              of the OLDBAB M.C. program of                       *
*              F.A. BERENDS AND R. KLEISS                          *
*              [1] NUCL. PHYS. B228 (1983) 537                     *
*              Changes were done by S. Jadach,                     *
*              E. Richter-Was and other                            *
*              authors of the paper                                *
*              [2] Phys. Lett. B253 (1991) 173                     *
*              All modifications are desribed                      *
*              in detail in the source code.                       *
*              PLEASE CITE references [1,2] !                      *
********************************************************************
********************************************************************
*              *********************************                   *
*              * This program is part of       *                   *
*              *   BHLUMI version 2.01         *                   *
*              *   September     1991          *                   *
*              *         AUTHORS               *                   *
*              * S. Jadach, E. Richter-Was     *                   *
*              *    B.F.L. Ward, Z. Was        *                   *
*              *********************************                   *
********************************************************************
********************************************************************
*                      OLDBIS input                                *
*           1011                option    switch       KEYOPT    N1 *
*              1                VARRAN    switch       KEYRND       *
*              1                weighting switch       KEYWGT       *
*              1                radiation switch       KEYRAD    N2 *
*              1                interf. s-chan.        KEYSIN       *
*              0                QM emiss.only/test     KEYMIR       *
*      92.50000000              CMSENE                 CMSENE    X1 *
*       1.70000000              THMIN                  THMIN     X2 *
*      17.06235297              THMAX                  THMAX     X3 *
*       0.00010000              XK0                    XK0       X4 *
*       0.99990000              XKMAX                  XKMAX     X5 *
*       0.00000000              XKMIN                  XKMIN     X6 *
********************************************************************
====================DUMPS====================
 P2    1.3965751180044   0.0000000000000  46.2289095473792  46.2500000000000
 Q2   -1.3965751180044   0.0000000000000 -46.2289095473792  46.2500000000000
SUM    0.0000000000000   0.0000000000000   0.0000000000000  92.5000000000000
====================DUMPS====================
 P2   -1.5625816325155   2.2484779230733  45.7232911219251  45.8052034745959
 Q2    1.5634040815780  -2.2492220660096 -46.1684491726400  46.2496370929672
PHO   -0.0008224490625   0.0007441429364   0.4451580507150   0.4451594324369
SUM    0.0000000000000   0.0000000000000   0.0000000000000  92.5000000000000
********************************************************************
*              *********************************                   *
*                     OUTPUT FROM                                  *
*                OLDBIS:        WINDOW A                           *
*              *********************************                   *
*                 X.sect. in [nb] units                            *
*          50000                generated events       NEVGEN    A0 *
*      -32.72299830  +--0.06495683  Xsec.     soft                 A1 *
*      167.59323787  +- 0.25328655  Xsec.     hard                 A2 *
*      134.93177133  +- 0.84285960  Xsec. straight       total     A3 *
*      134.87023957  +- 0.26148320  Xsec. clever         total     A4 *
```

```
*                  More on weights etc...                                      *
*     136.99681950                 crude Xs. soft                           A5 *
*     196.30528472                 crude Xs. hard                           A6 *
*      -0.23885955  +--0.00047415  aver. wt. soft                           A7 *
*       0.85373778  +- 0.00129027  aver. wt. hard                           A8 *
*     136.71830899                 Born  Xs.                                A9 *
********************************************************************************
********************************************************************************
*                 OLDBIS:          WINDOW B                                    *
*                    auxiliary information                                     *
*     134.87023957  +- 0.26148320  KSIN=1, no interf.        total     B1 *
*     134.88345088  +- 0.26154204  KSIN=2, with interf       total     B2 *
********************************************************************************
         LOWEST ORDER CROSS SECTION =   0.136718D+03 NB = UNIT
  APPROXIMATED HARD PHOTON XSECTION =       1.435838
         EXACT HARD PHOTON XSECTION =       1.225829
                        UNCERTAINTY =       0.001853
W.R.P EFFICIENCY IN INTERNAL C LOOP =       0.820781
 "      "        OF C/CT RESTRICTION =       0.982449
 "      "           FOR HARD WEIGHTS =       1.017865
  APPROXIMATED SOFT PHOTON XSECTION =       1.002037
         EXACT SOFT PHOTON XSECTION =      -0.239346
                        UNCERTAINTY =       0.000475
 W.R.P. EFFICIENCY FOR SOFT WEIGHTS =       1.000000
   APPROXIMATED TOTAL CROSS SECTION =       2.437875
          EXACT TOTAL CROSS SECTION =       0.986483
                        UNCERTAINTY =       0.001913
----------> TOTAL CROSS SECTION =====   0.134870D+03 NB
---------->  SOFT CROSS SECTION =====  -0.327230D+02 NB
---------->  HARD CROSS SECTION =====   0.167593D+03 NB
0 GENERATED WEIGHTS:
         < 0 IN HARD PART =        1.000000
      > WMAX IN HARD PART =        0.000000
     MINIMUM IN HARD PART =       -0.000005
     MAXIMUM IN HARD PART =        0.999675
         < 0 IN SOFT PART =    20543.000000
      > WMAX IN SOFT PART =        0.000000
     MINIMUM IN SOFT PART =       -0.515205
     MAXIMUM IN SOFT PART =        0.000000
================================================================================
=               ============ BOKER1 =============                              =
=               *********************************                              =
=               Integrated x-sect. various cuts                                =
=                     Any sub-generator                                        =
=                     PURE BREMSSTRAHLUNG                                      =
=               th.6118 fig.2a, middle bin                                     =
=               th.6118 fig.3,  middle bin                                     =
=               *********************************                              =
=                      ////// wide /////                                       =
=      43.59284971  +- 0.58155633  xsec. total             XS                  =
=       0.01334064                 rel. error              DS/XS               =
=      -0.06524725  +- 0.01247020  O(alfi)/Born -1         RXS                 =
=                     ///// narrow /////                                       =
=      24.55528827  +- 0.45453296  xsec. total             XS                  =
=       0.01851059                 rel. error              DS/XS               =
=      -0.07708437  +- 0.01708372  O(alfi)/Born -1         RXS                 =
=                      ////// mixed /////                                      =
=               th.6118 Tab.2a line.2, for BHLUM2                              =
=               th.6118 Tab.2b line.2 (!!!)                                    =
=      25.83844022  +- 0.32563756  xsec. total             XS                  =
=       0.01260283                 rel. error              DS/XS               =
=      -0.02885683  +- 0.01223916  O(alfi)/Born -1         RXS                 =
================================================================================
================================================================================
=               ============ BOKER2 =============                              =
=                     OLDBIS related tests                                     =
=               *********************************                              =
=               Integrated x-sect. no interf.                                  =
=               th.5888 tab.1, third column                                    =
=               *********************************                              =
=                      ////// wide /////                                       =
=      43.59284971  +- 0.58155633  xsec. total             XS                  =
=       0.01334064                 rel. error              DS/XS               =
=      -0.06524725  +- 0.01247020  O(alfi)/Born -1         RXS                 =
=                     ///// narrow /////                                       =
=      24.55528827  +- 0.45453296  xsec. total             XS                  =
=       0.01851059                 rel. error              DS/XS               =
=      -0.07708437  +- 0.01708372  O(alfi)/Born -1         RXS                 =
=                      ////// mixed /////                                      =
=      25.83844022  +- 0.32563756  xsec. total             XS                  =
=       0.01260283                 rel. error              DS/XS               =
=      -0.02885683  +- 0.01223916  O(alfi)/Born -1         RXS                 =
=               *********************************                              =
=                     UP-DOWN Interference                                     =
=               th.5888 tab.1, forth column                                    =
=               *********************************                              =
=                      ////// wide  /////                                      =
*       0.00011126  +- 0.00003452  O(alfi-alfj)/Born       RXS                 *
=                      ///// narrow /////                                      =
*       0.00016825  +- 0.00003669  O(alfi-alfj)/Born       RXS                 *
=                      ////// mixed /////                                      =
=               th.6118 tab.2b line.5                                          =
*       0.00016553  +- 0.00002618  O(alfi-alfj)/Born       RXS                 *
================================================================================
```