The Monte Carlo Event Generator

**YFSWW3 version 1.16**

for **W-Pair Production and Decay**

at **LEP2/LC Energies**†

S. Jadach\(^{a,b,c}\), W. Płaczek\(^{d,c}\), M. Skrzypek\(^{b,c}\) B.F.L. Ward\(^{e,c,f}\) and Z. Was\(^{b,c}\)

\(^{a}\)DESY Zeuthen, Platanenallee 6, D-15738 Zeuthen, Germany

\(^{b}\)Institute of Nuclear Physics, ul. Kawiory 26a, 30-055 Cracow, Poland,

\(^{c}\)CERN, CH-1211 Geneva 23, Switzerland,

\(^{d}\)Institute of Computer Science, Jagellonian University, ul. Nawojki 11, 30-072 Cracow, Poland,

\(^{e}\)Department of Physics and Astronomy,

The University of Tennessee, Knoxville, TN 37996-1200, USA,

\(^{f}\)SLAC, Stanford University, Stanford, CA 94309, USA.

Abstract

We present the Monte Carlo event generator **YFSWW3 version 1.16** for the process of **W-pair production and decay** in electron–positron collisions. It includes \(\mathcal{O}(\alpha)\) electroweak radiative corrections in the **WW** production stage together with the \(\mathcal{O}(\alpha^3)\) initial-state-radiation (ISR) corrections in the leading-logarithmic (LL) approximation, implemented within the Yennie–Frautschi–Suura (YFS) exclusive exponentiation framework. The photon radiation in the **W** decays is generated by the dedicated program **PHOTOS** up to \(\mathcal{O}(\alpha^2)\) LL, normalized to the **W** branching ratios. The program is interfaced with the \(\tau\) decay library **TAUOLA** and the quark fragmentation/hadronization package **JETSET**. The semi-analytical code **KorWan** for the calculations of the differential and total cross-sections at the Born level and in the ISR approximation is included.

To be submitted to *Computer Physics Communications*

† Work partly supported by the Polish Government grant KBN 5P03B12420, the European Commission 5-th framework contract HPRN-CT-2000-00149, the US DoE Contracts DE-FG05-91ER40627 and DE-AC03-76ER000515, and the Polish–French Collaboration within IN2P3 through LAPP Annecy.

CERN-TH/2001-017

DESY 01-027

January 2001
# Contents

1. **Introduction**  
2. **Physics Contents of YFSWW3**
   2.1 Notation  
   2.2 Master Formula  
   2.2.1 YFS Infrared Factors  
   2.2.2 Coulomb Correction  
   2.2.3 Anomalous Triple-Gauge Couplings  
   2.2.4 $\beta$'s for $W$-Pair Production  
   2.2.5 Prettublated EW Corrections  
   2.2.6 Fixed-Order Calculation  
   2.2.7 Radiative Corrections for $W$ Decays  
   2.2.8 ISR Reference Differential Distribution  
   2.3 YFS Reduction/Extrapolation Procedures  
   2.4 Multichannel YFS Reduction/Extrapolation  
3. **Monte Carlo Algorithm**  
   3.1 Main Steps in the Construction of the MC Weight  
   3.2 MC Weights and Absolute Normalization  
4. **Structure of the Program**  
   4.1 demo - Demonstration Program  
   4.2 data_files - Default Input Data  
   4.3 yfsww - Master Unit  
   4.4 model - Matrix Elements  
   4.5 ewc - Electroweak Library  
   4.6 interfs - Interfaces to External Libraries  
   4.7 wdeclib - External Libraries for $W$ Decay  
   4.8 glib - Histogramming and Numerical Library  
   4.9 semian - Semi-Analytical Program KorWan  
   4.10 ff - FF Package  
   4.11 dok - Related Papers  
   4.12 rew - Reweighting Tools  
   4.12.1 Reweighting Korali Events by YFSWW3  
   4.12.2 Reweighting YFSWW3 Events by YFSWW3  
   4.12.3 Reweighting YFSWW3 Events by Korali  
5. **How to use the Program**  
   5.1 Principal Entries of YFSWW3  
   5.2 Input/Output Parameters  
   5.3 Printouts of the Program
6 Summary 35
A Program Parameters and Their Settings 37
B Output of the Demo Program 44
PROGRAM SUMMARY

Title of the program: YFSWW3, version 1.16

Computer: any with the FORTRAN 77 compiler and the UNIX/Linux operating system

Operating system: UNIX/Linux

Programming language used: FORTRAN 77

High speed storage required: < 10 MB

Size of distributed program: 3,237 kB

Distribution format: tar gzip file (size of 1,103 kB)

No. of cards in combined program and test deck: about 23,000 plus about 48,000 of auxiliary packages: KorWan, FF, JETSET, TAUOLA, PHOTOS.

Keywords:
Standard Model (SM), LEP2, linear colliders (LC), quantum electrodynamics (QED), quantum chromodynamics (QCD), boson W, W-pair production, W decay, W branching ratio (BR), triple gauge boson couplings (TGC), quartic gauge boson couplings (QGC), four-fermion (4f) background, radiative corrections, Yennie–Frautschi–Suura (YFS) exponentiation, initial-state radiation (ISR), leading-log (LL) approximation, Coulomb effect, final-state radiation (FSR), electroweak (EW) corrections, leading-pole approximation (LPA), Monte Carlo (MC) simulation/generation.

Nature of the physical problem:
The process of the W-pair production is important for precise tests of the Standard Model as well as searches for “new physics” at LEP2 and future linear colliders. In order to match the experimental precision necessary for a successful physics programme, quantum effects (the so-called radiative corrections) have to be included into a theoretical description of this process. It turns out that not only the so-called universal corrections (initial-state radiation, the Coulomb effect, “naive” QCD corrections, etc.) are necessary, but also the $\mathcal{O}(\alpha)$ electroweak corrections in the $WW$ production are needed to reach the desired theoretical accuracy. All these effects should, preferably, be included in a Monte Carlo event generator in order to account for realistic experimental set-ups.

Method of solution:
The Monte Carlo event generator for the combined $W$-pair production and decay process including $\mathcal{O}(\alpha^3)$ LL ISR effects, the Coulomb correction (usual or screened), the “naive” QCD effect, the $\mathcal{O}(\alpha)$ EW corrections in the $WW$ production stage, implemented within the YFS exclusive exponentiation framework is provided. Multiphoton radiation in the $WW$ production is generated according to the YFS MC method. The photon radiation in the $W$ decays, normalized to the $W$ BRs, is generated by the LL-type MC program PHOTOS (up to two photons). The decays of $\tau$’s including radiative corrections are simulated by the dedicated package TAUOLA. The quark fragmentation/hadronization is performed with the help of the Lund program JETSET. The program can provide both weighted and
unweighted (weight = 1) events. Any experimental cut and apparatus efficiency may be introduced easily by rejecting some of the generated events.

Restrictions on the complexity of the problem:
The LPA is used (in two versions) to describe the signal $WW$ production process. Multiphoton radiation according to the YFS exponentiation scheme is generated only for the $WW$ production stage. $\mathcal{O}(\alpha)$ EW corrections (in LPA) are included only in the $WW$ production stage. The ISR effects beyond $\mathcal{O}(\alpha)$ are included in the LL approximation. Non-factorizable corrections (interferences between the production and decay stages) are approximated by the so-called screened Coulomb ansatz. Spin correlations between the $WW$ production and decays are fully included only at the Born and the ISR levels. Radiative corrections in the $W$ decays are included into an overall normalization through the $W$ BRs, and the real photon radiation is generated in the LL approximation (up to two photons) by the program PHOTOS. Anomalous triple gauge boson couplings are included in the Born-like matrix element, i.e. with the universal SM corrections only. Quartic gauge boson couplings are implemented according to the Standard Model only (no anomalous couplings). The $\tau$ decays and quark hadronization are performed, respectively, with the help of the dedicated packages TAUOLA and JETSET. No 4f background processes are included.

Typical running time:
200 CPU seconds of a PC Intel Pentium III @ 550MHz per 1000 unweighted events, for the parameter settings as they are given in the demonstration program.
1 Introduction

The process of $W$-pair production in electron–positron colliders is very important for testing the Standard Model (SM) and searching for signals of possible “new physics”; see e.g. Refs. [1,2]. One of the main goals when investigating this process at present and future $e^+e^-$ experiments is to measure precisely the basic properties of the $W$ boson, such as its mass $M_W$ and width $\Gamma_W$. This process also allows for a study of the triple and quartic gauge boson couplings at the tree level, where small deviations from the subtle SM gauge cancellations can lead to significant effects on physical observables – these can be signals of “new physics”.

In this work we present a Monte Carlo (MC) event generator that simulates the production and decay of the $W$ pair at $e^\pm$ colliders. The integrated cross sections and arbitrary differential distributions can be calculated from a series of constant-weight fully inclusive MC events. The program embodies the Standard Model scattering matrix element with the $\mathcal{O}(\alpha)$ radiative corrections for the doubly-resonant component of the scattering matrix element. The basics of the model used in the program and selected numerical results were already presented in earlier publications [3–6]. The important components of the presented program/calculation are the complete $\mathcal{O}(\alpha)$ radiative correction for on-shell $W$-pair production, which are implemented according to Refs. [7–10], see also Ref. [11].

The present YFSWW3 MC event generator was instrumental in achieving the new improved theoretical precision of 0.4% for the total cross section of the $WW$ production process at the highest LEP2 energies, as a result of direct and detailed comparison with the RACOONWW MC program [12, 13] within the 2000 LEP2 MC workshop (this conclusion was also supported by the comparison between RACOONWW and the calculation of Ref. [14]), see Section 4.3 in Ref. [2].

Let us briefly discuss the basic physics properties of the $W$-pair production and decay process. Since the $W$’s are unstable and short-lived particles, the $W$ pairs are not observed directly in the experiments but through their decay products: four-fermion ($4f$) final states (which may then also decay, radiate gluons/photons, hadronize, etc.). As high energy charged particles are involved in the process, one can also observe energetic radiative photons. So, at the parton level, one has to consider a general process:

$$e^+ + e^- \rightarrow 4f + n\gamma, \ (n = 0, 1, 2, \ldots),$$

where also some background (non-$WW$) processes contribute. In a theoretical description of this process – according to quantum field theory – one also has to include virtual effects, the so-called loop corrections. This general process is very complicated since it involves $\sim 80$ different channels ($4f$ final states) with complex peaking behavior in multiparticle phase space and a large number of Feynman diagrams. Even in the massless-fermion approximation, the number of Feynman graphs grows up from 9–56 per channel at the Born level to an enormous 3579–15948 at the one-loop level [14]. The full one-loop calculations

\footnote{We are grateful to the authors of these works for providing us with the relevant parts of the computer code.}
have not been finished yet, even for the simplest case (doubly plus singly $W$-resonant diagrams) \[16\]. But even if they existed one would be faced with problems in their numerical evaluation in practical applications, particularly within Monte Carlo event generators – they would be prohibitively huge and slow. These are the reasons why efficient approximations in the theoretical description of this process are necessary. These approximations should be such that on the one hand they would include all contributions/corrections that are necessary for the required theoretical accuracy (dependent on the experimental precision) and on the other hand they would be efficient enough for numerical computations. Given the complicated topologies of the $(4f + n\gamma)$ final states, such calculations should be, preferably, given in terms of a Monte Carlo event generator that would allow one to simulate the process directly \[17, 18\].

Our solution to this consists of two complementary Monte Carlo event generators: \texttt{YFSWW3} and \texttt{KoralW}. The latter includes the full lowest-order $e^+e^- \to 4f$ process, but with simplified radiative corrections – the universal ones such as initial-state radiation (ISR), the Coulomb effect, etc. In \texttt{YFSWW3}, on the other hand, the lowest-order process is simplified – only the doubly $W$-resonant contributions are taken into account, but inclusion of the radiative corrections in this process goes beyond the universal ones. In the current version of \texttt{YFSWW3} only those non-universal (non-leading) corrections are included that are necessary to achieve the theoretical precision for the total $WW$ cross section of 0.5% required for LEP2. In order to achieve a gauge-invariant description of the signal $WW$ process and to employ the existing $O(\alpha)$ electroweak corrections for the on-shell $W$-pair production, we use the so-called leading-pole approximation (LPA). A double-pole variant of the LPA (usually referred to as DPA) was advocated in Ref. \[19\] and applied in a variety of calculations, see for example Refs. \[6, 12–14, 20\] and Ref. \[2\] for a review and further references. The important thing is that \texttt{YFSWW3} and \texttt{KoralW} programs have a well established common part, which is the doubly $W$-resonant ($WW$) process with the same universal radiative corrections. This allows us to combine the results of the two programs so as to achieve the desired theoretical precision for $WW$ observables. Detailed descriptions of \texttt{KoralW} were published in Refs. \[21–24\]. Here, we present a documentation of the program \texttt{YFSWW3}; physical aspects together with some numerical results are discussed in Refs. \[3, 4, 20\]. Possible scenarios of combining the results of the two programs are presented in Refs. \[3, 20, 25\].

The outline of the paper is as follows. In Section 2, we define our notation and describe the physics contents of the program. In Section 3, we discuss the MC algorithm. In Section 4, we present the structure of the program, important routines, etc. Details about the practical use of the program are given in Section 5. Finally, Section 6 summarizes the paper. Appendices contain useful technical information on the structure of the program, its input/output, etc.
2 Physics Contents of YFSWW3

In this section we describe we define differential distributions used to calculate the cross section and generate events for the $W$-pair production and decay process. These distributions we define quite completely, because they were not defined in every detail in the past publications. On the other hand we shall skip the detailed discussion of the physics models that they represent. This will be done in a separate publication [26]. First, we discuss our notation. Then we present the master formula and its ingredients. And finally, we give some details on the MC algorithms implemented in the program.

2.1 Notation

![Diagram of W-pair production and decay process with multiple photons in the production stage.]

In YFSWW3 we adopt a notational convention similar to the one used in KoralW. The variables $p_1$ and $p_2$ are the incoming $e^-$ and $e^+$ four-momenta, respectively:

\[ p_1 = (E_e, 0, 0, p_e), \quad p_2 = (E_e, 0, 0, -p_e), \]  

where $E_e$ is the beam energy and $p_e = \sqrt{E_e^2 - m_e^2}$, with the electron mass $m_e$. The variables $Q_1$ and $Q_2$ are the four-momenta of the $W^-$ and $W^+$ bosons; their invariant masses are denoted by $M_1$ and $M_2$, respectively; $q_1, q_2$ and $m_1, m_2$ are the four-momenta and masses of the $W^-$ decay products; $q_3, q_4$ and $m_3, m_4$ are the analogous four-momenta and masses of the $W^+$ decay products, with the first component of each pair corresponding to a fermion and the second to an antifermion. By $k_i$ we denote the four-momenta of
radiative photons. See also fig. for a pictorial representation. The components of a four-vector are denoted by \( p = (p_0, p_x, p_y, p_z) \), with the metric convention of \( \{1, -1, -1, -1\} \). We also use the following Lorentz invariants: \( s = (p_1 + p_2)^2 \), \( s_1 = Q_1^2 = M_1^2 \) and \( s_2 = Q_2^2 = M_2^2 \).

### 2.2 Master Formula

Here, we consider the process:

\[
e^{-} + e^{+} \rightarrow \{W^{-} \rightarrow f_1 + \bar{f}_2\} + \{W^{+} \rightarrow f_3 + \bar{f}_4\} + n\gamma, \quad (n = 0, 1, 2, \ldots),
\]

where the radiative photons are emitted only at the \( WW \)-production stage. The photon radiation in \( W \) decays (FSR) in this version of \( \text{YFSWW3} \) is done, in the \( \mathcal{O}(\alpha^2) \) leading-logarithmic (LL) approximation, by the dedicated program \( \text{PHOTOS} \) \(^{[27]} \) — in the same way as in \( \text{KoralW} \). Since this is done externally on fully generated events (it does not change event weights), we do not include the FSR in our discussion of the master formula. It should be remembered that the use of \( \text{PHOTOS} \) in \( \text{YFSWW3} \) is a temporary solution before the advent of the YFS-exponentiated MC for \( W \) decays.

As was already mentioned in the Introduction, we use the LPA to describe this process respecting the \( SU(2)_L \times U(1) \) gauge invariance. In \( \text{YFSWW3} \), we implemented two options of the LPA, called LPA\(_a\) and LPA\(_b\). The first one, which is the default (recommended) option, is based on the approach of Stuart \(^{[19]} \), where only the Lorentz scalar functions in the S-matrix are expanded about poles corresponding to unstable \( W \)'s. In the lowest order, the LPA\(_a\) matrix element coincides with the so-called CC03 (see e.g. Ref. \(^{[15]} \) for an explanation of this notation) matrix element in the ’t Hooft–Feynman gauge. In the following we shall use the notation “CC03” in that sense. The second option, intended for some dedicated tests, follows the method suggested in Ref. \(^{[15]} \), where the whole S-matrix residuals are expanded about poles \(^{[1]} \); see Refs. \(^{[6,20]} \) for more details. In the following we shall describe the LPA\(_a\) scheme explicitly; LPA\(_b\) can be obtained from this by taking the \( W \)'s on shell (i.e. \( Q_1^2, Q_2^2 \rightarrow M_W^2 \)) everywhere, except for the Breit–Wigner denominators and the phase-space integration.

In the framework of the YFS exclusive exponentiation (EEX), the cross section for the process \(^{[3]} \) can be written as

\[
\sigma_{\text{Best}} = \sum_{n=0}^{\infty} \frac{1}{n!} \int ds_1 ds_2 \frac{d^3 Q_1}{Q_1^0} \frac{d^3 Q_2}{Q_2^0} \prod_{l=1}^{4} \frac{d^3 q_l}{q_l^0} \prod_{i=1}^{n} \frac{d^3 k_i}{k_i^0} \delta^{(4)}(p_1 + p_2 - Q_1 - Q_2 - \sum_{i=1}^{n} k_i) \delta^{(4)}(Q_1 - q_1 - q_2) \delta^{(4)}(Q_2 - q_3 - q_4) \rho_{\text{Best}}^{n}(p_1, p_2; q_1, \ldots, q_4, k_1, \ldots, k_n),
\]

\(^{2}\)We do not apply, however, the pole approximation to the phase space, as was done in the semi-analytical calculations of Ref. \(^{[14]} \).
where $\rho_n^{\text{Best}}$ is the following fully exclusive multiphoton differential distribution:

$$\rho_n^{\text{Best}}(p_1, p_2; q_1, \ldots, q_4, k_1, \ldots, k_n) = \prod_{i=1}^{n} \tilde{S}(p_1, p_2, Q_1, Q_2, k_i) \theta(k_i^0 - k_i)$$

$$\times e^{Y'(p_1, p_2, Q_1, Q_2, k_i)} [1 + \delta_c(Q_1, Q_2, M_W, \Gamma_W)] \sum_{\varphi \in \{I, W\}^n} p_\varphi B\{\{p, Q, k\}^{\varphi}\},$$

(5)

where $B$ is the contribution from the infrared-finite YFS residuals:

$$B\{\{p, Q, k\}^{\varphi}\} = \left[ 1 + \delta^{\text{TGC}}_{\text{An}}\{\{p, Q, k\}^{\varphi}\} \right]$$

$$\times \left\{ \tilde{\beta}_0^{(1)}\{\{p, Q, k\}^{\varphi}\} + \sum_{i=1}^{n} \tilde{\beta}_1^{(1)}\{\{p, Q, k_i\}^{\varphi}\} + \Delta \tilde{\beta}_0^{(3)}\{\{p, Q, k_i\}^{\varphi}\} \right. $$

$$+ \left. \sum_{\varphi_{i,j}=I} \frac{\Delta \tilde{\beta}_2^{(3)}\{\{p, Q, k_i, k_j\}^{\varphi}\}}{S_I(\{p\}^{\varphi}, k_i)} + \frac{1}{2!} \sum_{\varphi_{i,j}=I} \frac{\tilde{\beta}_3^{(3)}\{\{p, Q, k_i, k_j, k_l\}^{\varphi}\}}{S_I(\{p\}^{\varphi}, k_i) S_I(\{p\}^{\varphi}, k_j) S_I(\{p\}^{\varphi}, k_l)} \right\},$$

(6)

where we use a short-hand notation of the type $\{p, Q, k\} \equiv \{p_1, p_2, Q_1, Q_2, q_1, q_2, q_3, q_4\}$.

The sum $\sum_\varphi$ in Eq. (5) runs over all possible photon associations to the initial-state radiation (ISR) or the $W$-state radiation (WSR). For $n$ photons we identify each of the $2^n$ partitions $\varphi$ with a vector $\varphi = (\varphi_1, \ldots, \varphi_n)$, where $\varphi_i = I, W$ ($I$ and $W$ denote the ISR and the WSR photons, respectively). The partition weight $p_\varphi$ is given by

$$p_\varphi = \prod_{\varphi_{i}=I} \tilde{S}_I(k_i) \prod_{\varphi_{j}=W} \tilde{S}_W(k_j) = \prod_{\varphi_{i}=I} \tilde{S}_I(k_i) \prod_{\varphi_{j}=W} \tilde{S}_W(k_j)$$

$$\sum_\varphi \prod_{\varphi_{i}=I} \tilde{S}_I(k_i) \prod_{\varphi_{j}=W} \tilde{S}_W(k_j) = \prod_{i=1}^{n} \left( \tilde{S}_I(k_i) + \tilde{S}_W(k_i) \right),$$

(7)

where $\tilde{S}_I(k)$ and $\tilde{S}_W(k)$ are the real photon infrared (IR) factors for the ISR and the WSR, respectively:

$$\tilde{S}_I(p_1, p_2, k) = -\frac{\alpha}{4\pi} \left( \frac{p_1}{kp_1} - \frac{p_2}{kp_2} \right)^2,$$

$$\tilde{S}_W(Q_1, Q_2, k) = -\frac{\alpha}{4\pi} \left( \frac{Q_1}{kQ_1} - \frac{Q_2}{kQ_2} \right)^2.$$

(8)

It is worthwhile to stress that the partition dependence for the first two terms proportional to $\tilde{\beta}_0^{(1)}$ and $\tilde{\beta}_1^{(1)}$ cancels out, when truncated to $O(\alpha)$ (as it should), thanks to the special

---

3In sums like $\sum_{\varphi_{i,j}=I}$ we include all pairs of indices $(i, j)$, so that $i \neq j, \varphi_i = I$ and $\varphi_j = I$.  

---
normalization choice for the partition weights: \( \sum_{\wp \in \{I,W\}} n_{\wp} = 1 \). See also the definition of \( \bar{\beta}_1^{(1)} \) and the discussion on the partition-dependent “reduction procedure” \( R_\wp \) in the following. The purpose of the sum over partitions is to help the introduction of the LL corrections beyond the \( O(\alpha) \).

In Eq. (6) the two components proportional to \( \bar{\beta}_0^{(1)} \) and \( \bar{\beta}_1^{(1)} \) explicitly represent the complete \( O(\alpha) \) contribution to the \( WW \)-production process, while the remaining \( \bar{\beta}'s \) and \( \Delta \bar{\beta}'s \) collect the second- and third-order ISR corrections in the LL approximation; see Section 2.2.4 for more details. The dependence of the \( \bar{\beta} \) functions on the photon partitions enters through the so-called reduction (extrapolation) procedures, which take into account the CMS energy shift induced by the ISR; for more details see Subsection 2.2.4.

### 2.2.1 YFS Infrared Factors

The real photon IR factor

\[
\tilde{S}(p_1, p_2, Q_1, Q_2, k) = -\frac{\alpha}{4\pi^2} \left( \frac{p_1}{kp_1} - \frac{p_2}{kp_2} - \frac{Q_1}{kQ_1} + \frac{Q_2}{kQ_2} \right)^2
\]

includes the ISR, the WSR, and the interferences between them.

The step function \( \theta(k^0 - k_\epsilon) \), where \( k_\epsilon = \epsilon \sqrt{s}/2 \), cuts out the singular IR region, already included to all orders in the YFS form factor \( e_{Y'} \), where

\[
Y'(p_1, p_2, Q_1, Q_2; k_\epsilon) = 2\alpha \Re B'(p_1, p_2, Q_1, Q_2) + 2\alpha \tilde{B}(p_1, p_2, Q_1, Q_2; k_\epsilon).
\]

The IR functions \( \Re B' \) and \( \tilde{B} \) correspond to virtual and real photons, respectively, and are defined explicitly in Ref. [3].

### 2.2.2 Coulomb Correction

The correction \( \delta_C \) is the Coulomb effect, included both (two options) in its standard form as given in Ref. [28], and in terms of the so-called screened Coulomb ansatz of Ref. [29] which is an efficient approximation of the non-factorizable corrections of Refs. [30–33] for singly-inclusive distributions. It reads

\[
\delta_{C\{\text{Std,Scr}\}}(Q_1, Q_2, M_W, \Gamma_W) = \frac{\alpha\pi}{2\beta} \left[ 1 - \{1, (1 - \beta)^2\} \frac{2}{\pi} \arctan \left( \frac{|\kappa|^2 - p^2}{2p\Re \kappa} \right) \right],
\]

with \( \beta = \sqrt{1 - 2(s_1 + s_2)/s_Q + [(s_1 - s_2)/s_Q]^2} \), \( p = \frac{1}{2} \beta \sqrt{s_Q} \), \( s_Q = (Q_1 + Q_2)^2 \),

\[
\kappa = \sqrt{\frac{M_W}{2} \left( \sqrt{E^2 + \Gamma_W^2} - E \right) - i \sqrt{\frac{M_W}{2} \left( \sqrt{E^2 + \Gamma_W^2} + E \right)}, \quad E = \frac{s_Q - 4M_W^2}{4M_W}.\]

where $M_W$ and $\Gamma_W$ are the on-shell $W$ mass and $W$ width, respectively. The first term in the curly brackets in Eq. (11) corresponds to the standard (Std) correction, the second one to the screened (Scr) Coulomb correction. This correction is properly matched with the YFS form factor, where the Coulomb-like singularity appears in the virtual IR function $\Re B$. After matching it with the full Coulomb correction $\delta C$, the original YFS function $\Re B$ is replaced by $\Re B'$, as described in Ref. [3].

### 2.2.3 Anomalous Triple-Gauge Couplings

The anomalous corrections to the triple-gauge couplings (TGC) $WWV$ ($V = Z, \gamma$) are included multiplicatively through

$$1 + \delta_{\text{TGC}}^{\text{An}} = \frac{|M_{\text{Born}}^{\text{AnTGC}}|^2}{|M_{\text{Born}}^{\text{SM}}|^2},$$

where $M_{\text{Born}}^{\text{AnTGC}}$ is the Born-level matrix element with the anomalous TGCs, while $M_{\text{Born}}^{\text{SM}}$ is the corresponding SM matrix element. In YFSWW3 we use the same conventions (routines) for the TGCs as they are given in KoralW [23]. The anomalous couplings are included in the event generation only if an appropriate switch in the input parameters is on and the parameters corresponding to these couplings (in a given parameterization) are set up to non-SM values; see Appendix A for details.

### 2.2.4 $\bar{\beta}$’s for $W$-Pair Production

In the IR-finite $\bar{\beta}$ functions of Eq. (5), we see the superscripts $R_\nu$ in the four-momenta arguments like $\{p, Q, q\}^{R_\nu}$ or $\{p, Q, q, k\}^{R_\nu}$. This means that these arguments are subject to the “reduction (extrapolation) procedures”. These reduction procedures are partition-dependent – this is why they are denoted with the subscripts $\nu$. The partition dependence enters through accounting for the ISR energy shift $s \rightarrow s'$, where $s' = (p_1 + p_2 - \sum_{i=1}^{\nu} k_i)^2$ in the evaluation of the $\bar{\beta}$ functions. For more discussion on the meaning of the reduction/extrapolation procedures, see Subsection 2.3. In YFSWW3, we use the reduction procedures of the YFS2 and YFS3 programs of KORALZ [34–36], with some modifications: in constructing the reduced $W$ four-momenta, we take into account that, in general, $M_1 \neq M_2$. We perform appropriate transformations of the $W^\pm$ decay products, that is we boost them to $W$’s rest frames before the reduction, and then make the inverse boost transformation to the reduced frame. In general, the effect (recoil) of the CMS energy shift $s \rightarrow s'$ is taken due to the ISR photons only. This procedure is well justified for the ISR corrections beyond $O(\alpha)$ where we take $\{p, Q, q\}^{R_\nu}$ and for each photon we may say whether it belongs to the ISR or to the WSR. For the $O(\alpha)$ specific corrections we cannot make a distinction, in the corresponding $\bar{\beta}$’s, between the ISR and the WSR; the $\{p, Q, q, k_i\}^{R_\nu}$ in their arguments means that (i) the reduced $\{p, Q, k_i\}^{R_\nu}$ are within the 3-body phase space and (ii) the photon $k_i$ is excluded from the energy shift $s \rightarrow s'$ calculation. For more discussion on this non-trivial point see Subsections 2.3 and 2.4.
Let us now describe in detail the $\mathcal{O}(\alpha)$-complete functions $\beta^{(1)}_0$ and $\beta^{(1)}_1$. The first one, which includes the $\mathcal{O}(\alpha)$ electroweak (EW) virtual corrections, reads as follows:

$$\bar{\beta}^{(1)}_0\{p, Q\} = \bar{\beta}^{(0)}\{p, Q\} \left[1 + \delta^{(1)}_{\text{ISR}}(p_1, p_2)\right] + \bar{\beta}^{(0)}\{p, Q\} \left[\delta^{(1)}_{\text{v+s}}(p_1, k_e) - Y(p_1, Q; k_e) - \delta^{(1)}_{\text{ISR}}(p_1^\text{eff}, p_2^\text{eff})\right]_{G_{\mu}}$$

where $\beta^{(0)}_0$ is the lowest-order matrix element squared for the $W$-pair production and decay; it is identical to the corresponding $\beta^{(0)}_0$ function in KoralW $^{[21, 23]}$. In fact, the $\beta^{(1)}_0$ function comes in two variants. The first one, referred to as scheme (A) $^{[8]}$, reads as follows:

$$\bar{\beta}^{(1)}_{0, A}\{p, Q\} = \bar{\beta}^{(0)}\{p, Q\} \left[1 + \delta^{(1)}_{\text{ISR}}(p_1, p_2)\right] + \bar{\beta}^{(0)}\{p, Q\} \left[\delta^{(1)}_{\text{v+s}}(p_1, k_e) - Y(p_1, Q; k_e) - \delta^{(1)}_{\text{ISR}}(p_1^\text{eff}, p_2^\text{eff})\right]_{G_{\mu}}$$

where $\delta^{(1)}_{\text{ISR}} = (\alpha/\pi)\ln(s/m^2) - 1$ is the LL pure QED $\mathcal{O}(\alpha)$ ISR correction, and the subscript $G_{\mu}$ means that the whole correction is calculated in the so-called $G_{\mu}$ scheme $^{[7]}$ as explained in Ref. $^{[9]}$. Another option, referred to as scheme (B), similar to the one used in the program RACOONWW $^{[13]}$, is defined as follows:

$$\bar{\beta}^{(1)}_{0, B}\{p, Q\} = \bar{\beta}^{(0)}\{p, Q\} \left[1 + \delta^{(1)}_{\text{ISR}}(p_1, p_2)\right] + \bar{\beta}^{(0)}\{p, Q\} \left[\delta^{(1)}_{\text{v+s}}(p_1, k_e) - Y(p_1, Q; k_e) - \delta^{(1)}_{\text{ISR}}(p_1^\text{eff}, p_2^\text{eff})\right]_{G_{\mu}} \left(\frac{\alpha}{\alpha_{G_{\mu}}}\right)$$

where $\alpha$ is the fine structure constant and $\alpha_{G_{\mu}} = (\sqrt{2}G_{\mu}M_W^2\sin^2\theta_W)/\pi$. Here, although the $\mathcal{O}(\alpha)$ corrections are calculated in the $G_{\mu}$ scheme, their relative coefficient is $\alpha$ instead of $\alpha_{G_{\mu}}$. Our recommended option is scheme (A), which is inspired by the renormalization group equations (RGEs) $^{[9, 17]}$.

The $\mathcal{O}(\alpha)$ virtual and soft photon corrections for the $WW$ production stage are included in $\delta^{(1)}_{v+s}$. In the current version of the program, we use the calculations of Ref. $^{[4]}$ for the on-shell $WW$-production. Since in our case the $W$'s are off-shell, we have to perform some transformations/extrapolations in order to employ these calculations. We do it in the following way. After performing the reduction procedure for the $\tilde{\beta}_0$ function, as described above, we find $\cos \theta_{\text{eff}}$ and the velocity $\beta_{\text{eff}}$ of the $W$'s in the reduced frame, where

$$\beta_{\text{eff}} = \sqrt{1 - 2(s_1 + s_2)/s' + [(s_1 - s_2)/s']^2},$$
$$\cos \theta_{\text{eff}} = (\vec{p}_{1}^{R_{\nu}} \cdot \vec{Q}^{R_{\nu}})/(|\vec{p}_{1}^{R_{\nu}}||\vec{Q}^{R_{\nu}}|).$$

Then, for this value of $\beta_{\text{eff}}$ we calculate the effective CMS energy squared for on-shell $W$'s:

$$s_{\text{eff}} = 4M_W^2/(1 - \beta_{\text{eff}}^2).$$

$^4$In the early version of the present program called YFSWW2, corrections beyond the ISR were included only through the function $Y'$ and $S$-factors $^{[5]}$. 

12
Having \((s_{\text{eff}}, \beta_{\text{eff}}, \cos \theta_{\text{eff}})\), we construct the four-momenta \((p_{\text{eff}}^1, p_{\text{eff}}^2, Q_{\text{eff}}^1, Q_{\text{eff}}^2)\) for the on-shell process \(e^+ e^- \rightarrow W^+ W^-\) and use them to calculate the respective EW corrections. The same four-momenta are used to calculate the IR form factor \(Y\) according to Eq. (10); now the Coulomb-like singularity in the virtual IR function \(\Re B\) is kept, in order to cancel the corresponding singularity in \(\delta_{(1)}^{v+s}\), and to calculate the \(O(\alpha)\) ISR correction:

\[
\delta_{\text{ISR}}(p_{\text{eff}}^1, p_{\text{eff}}^2) = \frac{\alpha}{\pi} \left( \ln \frac{s_{\text{eff}}}{m_e^2} - 1 \right). \tag{19}
\]

Such a procedure allows us to use the EW corrections also in the \(WW\)-threshold region, however, in this region the LPA becomes less precise. Since there exists no other calculation with the \(O(\alpha)\) EW corrections for the \(W\)-pair production and decay process in the \(WW\)-threshold region, we estimate, for the moment, the precision of \textsc{YFSWW3} in that region at 2% conservatively, just as for the ISR approximation. This can be improved in the future, when more calculations/tests are done.

Two comments are in order about the approximations used in the evaluation of Eqs. (14–16):

1. In the LPA\(_{a,b}\), the one-loop corrections should be evaluated “on-pole”, i.e. for the complex \(W\) mass: \(M^2 = M_W^2 - i M_W \Gamma_W\). This would, however, require the analytic continuation of the on-shell one-loop corrections to the second Riemann sheet. We avoid this by using the approximation \(M \simeq M_W\) in the LPA residuals. Since this is done for the \(O(\alpha)\) corrections, the error introduced by this approximation is of the order of \((\alpha/\pi)(\Gamma_W/M_W) < 10^{-4}\), which is negligible for the precision we aim at.

2. A coherent sum over \(W\) spins between the production and decay stages is included in Eq. (14) in the \(\tilde{\beta}_0^{(0)}\) function, but not in the non-leading (NL) \(O(\alpha)\) EW correction. This should be sufficient for the LEP2 precision as the total EW correction in the LEP2 energy range is typically \(\sim 1–2\%\). However, the \(W\)-spin effects can be included also in the NL \(O(\alpha)\) EW correction, if necessary.

The \(O(\alpha)\) real photon ISR and photon emission in the \(WW\)-production stage are included through the following IR-finite function:

\[
\tilde{\beta}_1^{(1)} (\{p, Q, q, k\}^{R_\nu}) = D_{\gamma}^{(1)} (\{p, Q, q, k\}^{R_\nu}) - \tilde{S} (\{p, Q, q, k\}^{R_\nu}) \tilde{\beta}_0^{(0)} (\{p, Q, q\}^{R_\nu}), \tag{20}
\]

where \(D_{\gamma}^{(1)}\) is the differential cross section for the single hard photon radiation in the \(WW\)-production stage and the non-radiative \(W\)-decays. For this we use the calculations of Ref. [8], where the \(W\)’s are taken on shell. In order to accommodate them in our off-shell process, we use general formulae for \(W\)-polarization vectors in terms of their off-shell four-momenta [38] in the spin amplitude calculations, and perform a coherent summation over the \(W\)’s polarizations of the production and decay amplitudes. In fact, the hard-photon amplitudes of Ref. [8] are given in the massless-fermion approximation, so they become inaccurate when a photon is emitted close to a fermion direction. To correct
this, we supplement these amplitudes with the mass terms according to the CALKUL prescription of Ref. [39].

In Eqs. (4) and (5) we use the IR-finite $\beta$ functions for the ISR $\bar{\beta}_{0,\text{ISR}}^{(3)}$, $\bar{\beta}_{1,\text{ISR}}^{(3)}$, $\bar{\beta}_{2,\text{ISR}}^{(3)}$ and $\bar{\beta}_{3,\text{ISR}}^{(3)}$, calculated up to $O(\alpha^3)$ LL. They are exactly the same as in KoralW, see Ref. [23]. We define in addition:

$$
\Delta \bar{\beta}_{0,\text{ISR}}^{(3)} = \bar{\beta}_{0,\text{ISR}}^{(3)} - \bar{\beta}_{0,\text{ISR}}^{(1)}, \quad \Delta \bar{\beta}_{1,\text{ISR}}^{(3)} = \bar{\beta}_{1,\text{ISR}}^{(3)} - \bar{\beta}_{1,\text{ISR}}^{(1)}.
$$

In Eqs. (4) and (5), they are evaluated only for the ISR photons. In principle, these ISR residuals could be calculated for all photons – it is still consistent with the LL ISR approximation.

### 2.2.5 Pretabulated EW Corrections

We have implemented also the pretabulated (or “fast”) version of the EW corrections. In this mode, a pretabulation of the NL $O(\alpha)$ EW corrections

$$
\delta_{\text{EW}}^{\text{NL}}(\{p, Q\}) = \delta_{\text{v}}^{(1)}(\{p, Q\}; k_\epsilon) - Y(\{p, Q\}; k_\epsilon) - \delta_{\text{ISR}}^{(1)}(p_1, p_2)
$$

is done in the initialization stage of the program. Then, during the event generation, a linear interpolation of the pretabulated values is performed for each event. The pretabulation is done as a function of $\cos \theta$ of the $W^{-}$ for fixed values of the CMS energy $\sqrt{s}$ (set-up to the nominal energy) and of the $W$ invariant masses (set-up to $M_W$). This means, that in this approach, the ISR energy shift $s \to s'$ as well as the off-shellness of the $W$’s is not taken into account (the latter prevents this method from being used below the $WW$-threshold). In that respect, the pretabulated version of the EW corrections should be regarded as some approximation of the YFSWW3 “best” prediction. The main advantage of this version is that it is much faster in computing time than the exact (“best”) one, because here the time-consuming EW library is invoked only at the beginning of the program run for some number of equally spaced points instead of by calling it for each generated event. We have checked that at $\sqrt{s} = 200$ GeV, the results for the total cross section and main distributions of the pretabulated version differ from the corresponding results of the “best” one by only 0.1%, which is well below the required theoretical precision for the LEP2 data analysis.

### 2.2.6 Fixed-Order Calculation

By truncating the master formula (4) at some power of the coupling constant $\alpha$, one can obtain a fixed-order calculation, e.g. $O(\alpha^0)$, $O(\alpha^1)$.

At $O(\alpha^0)$ (the Born approximation), one gets the differential distribution

$$
\rho_0^{(0)}(\{p, Q, q\}) = \bar{\beta}_0^{(0)}(\{p, Q, q\}).
$$

At $O(\alpha^1)$, there are two contributions: (1) from the one-loop virtual corrections and the real soft photon radiation

$$
\rho_0^{(1)}(\{p, Q, q\}; k_\epsilon) = \bar{\beta}_0^{(0)}(\{p, Q, q\}) \left[ 1 + \delta_{v+\epsilon}^{(1)}(\{p, Q\}; k_\epsilon) \right],
$$

14
and (2) from the real hard photon radiation

$$\rho_1^{(1)}(p, Q, q, k) = D_\gamma^{(1)}(p, Q, q, k).$$

We have implemented the above fixed-order calculations in YFSWW3. They are available through the parallel weights, as explained in Table 9 of Appendix A. For $\delta_{\nu+\nu}^{(1)}$ we use the calculation of Ref. [7], and for $D_\gamma^{(1)}$ the calculation of Ref. [8]; see also Subsection 2.2.4 for more details.

### 2.2.7 Radiative Corrections for $W$ Decays

Radiative corrections in $W$ decays are not included in the above formulae explicitly. However, we include them in the overall normalization of the cross section by normalizing the $W$ decay amplitudes to the $W$ branching ratios (BR), which can be provided through the input parameters of the program. This means that in the $\beta$ functions of Eq. (6) for the $(i, j)$-th $W^-W^+$ decay channel, we use the normalization constant

$$N_{ij} = \frac{B_i B_j}{B_{ref}^2},$$

with

$$B_{ref} = \frac{\alpha_W M_W}{12 \Gamma_W \sin^2 \theta_W},$$

where $\theta_W$ is the weak mixing angle, and $\alpha_W = \alpha_{G_\mu}$ in our recommended $G_\mu$ scheme, while we set $\alpha_W = \alpha$ in the so-called $\alpha$ scheme. $B_i$ is the BR for the $i$-th $W$-decay channel, while $B_{ref}$ is the reference BR used for the normalization of the actual matrix element; see Ref. [21] for more details. For the fully inclusive process (all channels), the normalization constant is

$$N_{all} = \frac{1}{B_{ref}^2}.$$

It turns out that the BRs calculated at the Born level in the $G_\mu$ scheme – the so-called improved Born approximation (IBA) – agree within 0.1% with the full one-loop results [15]. Such an IBA option is also included in the program – for this one needs only to supply (through the input parameters) the values of the CKM matrix elements. In that case

$$B_i = \begin{cases} \frac{B_{ref}}{3 |V_{i}^{\text{CKM}}|^2} \left(1 + \frac{\alpha_s(M_W^2)}{\pi}\right) B_{ref} : & \text{for leptons,} \\ B_{ref} : & \text{for quarks,} \end{cases}$$

where $V_{i}^{\text{CKM}}$ is the CKM matrix element for the $i$-th $W$-decay channel, and $\alpha_s(M_W^2)$ is the QCD coupling constant at the $M_W$ scale.

By construction, the real photon radiation in $W$ decays, which is generated by PHOTOS, does not affect the overall normalization of the cross section – this is of course motivated by the KLN theorem [40, 41].
2.2.8 ISR Reference Differential Distribution

Equations (4) and (5) define the so-called “best” physics model of YFSWW3; if the corresponding weight is used as a rejection weight in order to generate unweighted (constant-weight) events, then we say that YFSWW3 is run in the “best” mode. We want to combine the fully exclusive (multiphoton) differential distributions from YFSWW3 and KoralW, so that the resulting distributions contain both the LPA $O(\alpha)$ EW corrections (missing in KoralW) and 4f-background diagram corrections (missing in YFSWW3), using event-per-event correction weights. In order to do this, we must define a certain simplified, auxiliary, fully exclusive, differential distribution $d\sigma_R$, which is common to both programs, and implement the corresponding MC weight $w_R$ in both programs. In YFSWW3 the weight $w_R$ can be used also optionally, as a main rejection weight – this is the so-called “ISR mode”. Otherwise, $w_R$ is one of many alternative weights calculated simultaneously with the “best model weight”; it is available to the user, and/or it is used to calculate the correction weight that serves the purpose of introducing the missing $O(\alpha)$ EW corrections in events generated by KoralW.

In the reference distribution $d\sigma_R$, only the numerically leading corrections, i.e. the ISR, the Coulomb correction, etc., are taken into account. This reference distribution includes the CC03 matrix element defined identically in YFSWW3 and KoralW. We also say, sometimes, that this is the distribution of the ISR-type model. The corresponding fully exclusive, reference, differential distribution, replacing $\rho_n$ of Eq. (4), we define as follows:

$$\rho_n^R(p_1, p_2; q_1, \ldots, q_n, k_1, \ldots, k_n) = \prod_{i=1}^n \bar{S}_I(p_1, p_2, k_i) \theta(k_i^0 - k_i) \times e^{Y_I(p_1, p_2; \omega)} \left[ 1 + \delta_C(Q_1, Q_2, M_W, \Gamma_W) \right] \left[ 1 + \delta^{TGC}_{\text{ISR}} \left( \{p, Q, q\}^R \right) \right]$$

$$\left\{ \sum_{i=1}^n \frac{\tilde{\beta}^{(3)}_{1,\text{ISR}} \left( \{p, Q, q\}^R, k_i \right)}{\bar{S}_I(\{p\}^R, k_i)} + \sum_{i>j}^n \frac{\tilde{\beta}^{(3)}_{2,\text{ISR}} \left( \{p, Q, q\}^R, k_i, k_j \right)}{\bar{S}_I(\{p\}^R, k_i)\bar{S}_I(\{p\}^R, k_j)} \right\},$$

where we have made the following replacements with respect to Eq. (4):

$$Y' \rightarrow Y_I(p_1, p_2; \omega) = \frac{2\alpha}{\pi} \left[ \ln \frac{s}{m_e^2} - 1 \right] \ln \epsilon + \frac{1}{2} \ln \frac{s}{m_e^2} - 1 + \frac{\pi^2}{2},$$

$$\bar{S} \rightarrow \bar{S}_I(p_1, p_2, k),$$

and we include the pure ISR LL $\tilde{\beta}$’s only. The WSR is just absent from the above distribution, which means that there is only one photon partition (where all photons are associated to the ISR) with the weight $p_n^R = 1$. In the actual MC program, $p_n^R$ should be implemented with a MC weight in the operational mode when the WSR is switched off – through the parallel weights. Both of these possibilities are included in YFSWW3. However,
it can also be implemented as a MC weight when the WSR is switched on. In such a case, the sums over the $\bar{\beta}$’s should extend to all photons.

The above ISR reference cross section is also used in this and other works to define the so-called non-leading (NL) corrections

$$\sigma_{NL} \equiv \sigma - \sigma_R = \int (d\sigma - d\sigma_R). \quad (32)$$

The subscript NL expresses the fact that these corrections are numerically smaller than the leading ISR corrections. It does not mean, however, that these corrections are free of logarithmic contributions, such as $\ln(s/M_W^2)$, but these logarithms are numerically much smaller than the ISR basic logarithm $\ln(s/m_e^2)$.

As already indicated, the reference ISR differential distribution of Eq. (30) (for the same CC03 matrix element) is also implemented in KoralW, and it has been checked numerically to a very high accuracy [6, 25] that its KoralW implementation agrees with that of the present YFSWW3 program. Having done that, it is possible to combine the results of the KoralW and YFSWW3 programs by reweighting the MC events, in order to include both the effects of the 4f background and the LPA $\mathcal{O}(\alpha)$ NL corrections in the MC predictions for the $W$-pair production and decay process. This is done in practice by reweighting the MC events produced by KoralW with the correction weight provided by YFSWW3. The correction $\delta_{NL}^{R}$ to be used by KoralW is defined as

$$1 + \delta_{NL}^{R} = \frac{\rho_{n}^{Best}(p_1, p_2; q_1, \ldots, q_4, k_1, \ldots, k_n)}{\rho_{n}^{R}(p_1, p_2; q_1, \ldots, q_4, k_1, \ldots, k_n)}, \quad (33)$$

where the numerator and denominators are defined in Eqs. (5) and (30). A few remarks are in due order. The above ratio is, in fact, calculated as the ratio of the corresponding MC weights. The YFSWW3 program has to be run in the mode with the WSR switched on, because $\rho_{n}^{Best}$ includes the summation over the ISR–WSR photon partitions, see Eq. (2), through the multichannel reduction procedure $\mathcal{R}_\mathcal{V}$. Under these conditions the above correction includes the complete $\mathcal{O}(\alpha)$ correction, which can be used to correct the fully exclusive distributions generated by KoralW, on an event-per-event basis. As we shall see in Subsection 4.12, the YFSW3 program provides at present only a numerically efficient approximation of the above $\delta_{NL}^{R}$.

### 2.3 YFS Reduction/Extrapolation Procedures

It is characteristic and inherent feature of the YFS exponentiation that the IR-finite $\bar{\beta}_i$ distributions require extrapolation to a larger phase space with the additional “spectator” photons. For instance, the function $\bar{\beta}_0$ is in our case originally defined at the $\{q_1, q_2, q_3, q_4\} = \{q\}$ point of the 4-particle phase space and it has to be also defined at every $\{q, k_1, \ldots, k_n\}$ point of the $(4 + n)$-particle phase space. We call the procedure of extending the domain of the $\bar{\beta}_0$ function $\bar{\beta}_0\{q\} \rightarrow \bar{\beta}_0\{q, k_1, \ldots, k_n\}$ the “extrapolation procedure”. The only true limitation in the choice of the extrapolation procedure is
that $\tilde{\beta}_0^E(\{q\},0,\ldots,0) = \tilde{\beta}_0(\{q\})$ must hold. Typically, $\tilde{\beta}_0(\{q\})$ is defined in terms of dot-products of the four-momenta $p_i \cdot q_j$ (or inner spinor products). Extrapolation may be done in a natural way by using exactly the same algebraic expressions in the larger phase space as were originally obtained from the Feynman diagrams in the smaller phase space. In the smaller phase space, the dot products obey certain relations due to the four-momentum conservation, which no longer hold true in the presence of the additional photons – this is the main mechanism in the realization of such an extrapolation. Another useful method of implementing the extrapolation (especially in the case when $\tilde{\beta}_0(\{q\})$ is implemented in the form of a black-box procedure, which requires as the input the four-momenta in the 4-particle phase space) is to project the $(\{q\}, k_1, \ldots, k_n)$ point into the “reduced” point $\{q\}^R$ using some kinematical manipulations (boosts, rotations, rescalings) on the four-momenta, and subsequently plugging it into $\tilde{\beta}_0^E(\{q\}, k_1, \ldots, k_n) \equiv \tilde{\beta}_0(\{q\}^R)$. The above “reduction procedure” is, of course, less general than the extrapolation procedure. Last, but not least, let us note that the extrapolation procedure $(\{q\}) \rightarrow (\{q\}, k_1, \ldots, k_n)$ is the reverse procedure to that of defining the residua at the IR-divergent points/poles $(\{q\}, k_1, \ldots, k_n) \rightarrow (\{q\}) = (\{q\}, 0, \ldots, 0)$ in the derivation of the resummation of the IR divergences to the infinite order (i.e. the derivation of the YFS exponentiation). The extrapolation procedure $\tilde{\beta}_1(\{q\}, k_j) \rightarrow \tilde{\beta}_1^E(\{q\}, k_1, \ldots, k_j, \ldots, k_n)$ is quite analogous. It has to obey the relation $\tilde{\beta}_1(\{q\}, k_j) \rightarrow \tilde{\beta}_1^E(\{q\}, 0, \ldots, 0, k_j, 0, \ldots, 0)$. Here all photons except the $j$-th one are the spectators.

It should be stressed that the above freedom, due to a reduction/extrapolation procedure, is a well-known feature of the YFS exponentiation, already underlined in Ref. [42] and discussed in many works implementing the YFS exponentiation; see for example Refs. [34, 43]. The uncertainty of the results due to the reduction/extrapolation procedure in the $\mathcal{O}(\alpha^n)$ YFS-exponentiated calculation is always at least of $\mathcal{O}(\alpha^{n+1})$; it does not influence or spoil the $\mathcal{O}(\alpha^n)$ perturbative “exact” contributions coming from the $\mathcal{O}(\alpha^n)$ Feynman-diagram calculations. On the contrary, a reasonable choice of the reduction/extrapolation procedure may improve significantly the total precision, because the YFS exponentiation is then able to sum up efficiently higher-order effects beyond the “exact” $\mathcal{O}(\alpha^n)$, as was often seen [44–47]. The above remark is also valid for the special “multichannel” variant of the reduction/extrapolation procedure described in the following.

### 2.4 Multichannel YFS Reduction/Extrapolation

It is useful sometimes to split the $\tilde{\beta}$’s into several components and introduce a different extrapolation/reduction procedure for each component. This is perfectly within the limits of our freedom. For instance in YFS3, in the centre of mass of the Z-boson, neither the initial beams nor the final fermion pairs have the momenta back-to-back. There are four possible definitions of the scattering angles in this frame. One defines $\tilde{\beta}_0$ as a sum of four differential Born-level distributions, each for a different scattering angle. They might be summed up with the same weight 1/4, but, in fact, also a more sophisticated mixture is implemented [48]. A similar multichannel extrapolation procedure is used in BHLUMI [43,45].
In the present work, we employ another kind of multichannel extrapolation, introduced for the first time in the unpublished program BHWIDE [46]. Let us elaborate on that method. Our exponentiation model is based on the real soft photon factor $\tilde{S}$ and the corresponding virtual photon function $\Re B$ to which photons emitted from the beams and from the $W$'s contribute coherently. We cannot, therefore, say whether a given photon is emitted from the initial state or from the $W$'s. We may, however, say something about it in a probabilistic way. If we split $\tilde{S}$ as follows:

$$\tilde{S}(k) = -\frac{\alpha}{4\pi^2} \left( \frac{p_1}{kp_1} - \frac{p_2}{kp_2} - \frac{Q_1}{kQ_1} + \frac{Q_2}{kQ_2} \right)^2 = \tilde{S}_I(k) + \tilde{S}_W(k) + \tilde{S}_{int}(k),$$

we may define for each photon a probability

$$p_I(k) = \frac{\tilde{S}_I(k)}{\tilde{S}_I(k) + \tilde{S}_W(k)} \tag{35}$$

that it was emitted from the beams, and the probability $p_W(k) = 1 - p_I(k)$ that it was emitted from the $W$'s. The above probabilities are a kind of LL concept, which has to be used with care. It may help to sum up higher-order QED corrections in the LL approximation. In particular, it may allow us to construct a multichannel type of reduction/extrapolation in which we may say for the spectator photons whether they were emitted by the beams (ISR) or by the $W$ bosons (WSR). This is done as follows: for each phase-space point we split the differential distributions into $2^n$ components, corresponding to $2^n$ possible associations of $n$ photons to the ISR or the WSR, the weight being the product $P(k)$ of the corresponding $p_I(k)$ or $p_W(k)$. This product is exactly the partition weight $p_\pi$ in Eqs. (4) and (7). For each component we may apply a separate reduction procedure which “knows” whether a given spectator photon belongs to the ISR or the WSR. The above procedure is, in practice, simpler than is said above, because in the MC realization we start from the YFS3 differential distribution, in which $\tilde{S}_{int}(k)$ is neglected, and the sum over the associations of photons to the ISR or the WSR is “randomized”. In other words, we generate only one of the $2^n$ possible associations and therefore, for a given event, we work with only one component (association), for which we may immediately employ its “native” reduction procedure – that is we may use the information on the photon associations directly from the YFS3 MC generator; see also the next section for an additional discussion. Let us stress again that this powerful solution should be used with care, especially for the non-spectator photons in the $\tilde{\beta}$ functions.

3 Monte Carlo Algorithm

In constructing the MC algorithm for the cross-section calculation and the event generation, we start from the master formula of Eq. (4) and perform step-by-step simplifications
in the differential distribution, compensating them with appropriate weights. We do this until we reach a simple enough differential distribution to be generated using the standard MC techniques. The MC algorithm of \textit{YFSWW3} for the Born-like $WW$ production and decay is identical to the one used in \textit{KoralW} for the so-called CC03 option \cite{21}, while the algorithm for the multiphoton radiation in the $e^+e^- \rightarrow W^+W^-$ process is based on the one implemented in the program \textit{YFS3} of \textit{KORALZ} \cite{36} for the process $e^+e^- \rightarrow f\bar{f}$. Since $W$'s are much heavier than light fermions and, in general, have different invariant masses, we had to make a few modifications in the latter algorithm. The most important of them are:

- generalization of the photon radiation formulae to allow for large and different masses of radiating particles\textsuperscript{3},
- inclusion of interferences between the radiation from the initial state and that from the $WW$ state, both in the YFS IR functions $\tilde{S}$ and $Y$ and in the IR-finite residuals $\beta$.

The algorithm of \textit{YFS3}, although used for a decade as part of other programs, was not documented until recently. Its first full description can be found in two recent works \cite{47,48}. We therefore do not describe it here in detail, but refer the interested reader to the above two papers. We explain below only the main steps in the \textit{YFSWW3} algorithm that lead to the algorithms of \textit{YFS3} and \textit{KoralW}, and then comment on the YFS reduction/extrapolation procedures.

### 3.1 Main Steps in the Construction of the MC Weight

The MC algorithm of \textit{YFSWW3} is based on the importance-sampling method. In constructing it, we start from the master formula (4) and make a series of simplifications compensating them with appropriate weights until we reach a distribution that is simple enough to be generated using the basic MC methods (see e.g. Refs. \cite{49,50}), as we indicated above. Here, we describe only the main simplifications in Eq. (4) that lead to the formula that can be generated with the help of the MC algorithms used in the programs \textit{YFS3} and \textit{KoralW}. They are:

1. $\tilde{\beta}$ functions:
   For the aggregate of the $\tilde{\beta}$ functions we do the following replacement:
   \begin{equation}
   B (\{p, Q, q, k\}^{R_\nu}) \rightarrow b_0 (\{p, Q, q\}^{R_\nu}),
   \end{equation}
   which is compensated by the “model” weight
   \begin{equation}
   w_{\tilde{\beta}}^{\nu} = \frac{B (\{p, Q, q, k\}^{R_\nu})}{b_0 (\{p, Q, q\}^{R_\nu})},
   \end{equation}
\textsuperscript{3}This extension is also included in the \textit{YFS3} of the \textit{K\&K} MC program \cite{48}.
where \( b_0 = \tilde{\beta}_0^{(0)} \) is the Born-level differential cross section for the \( W \)-pair production and decay process as given in Eq. (4) of Ref. [21]. To evaluate the model weight \( w_\beta \), we use the partition-dependent reduction procedures described in Subsection 2.2.4.

2. YFS form factor:
We simplify this with the replacement:

\[
Y'(p_1, p_2, Q_1, Q_2; k_e) \rightarrow Y_I(p_1, p_2; \epsilon) + Y_W(Q_1, Q_2; k_e),
\]

which is compensated by the weight

\[
w_Y = \exp[Y' - Y_I - Y_W],
\]

where \( Y_I \) is given in Eq. (31) and \( Y_W \) is the YFS IR function corresponding to the photon radiation from the \( W^-W^+ \) electric dipole:

\[
Y_W(Q_1, Q_2; k_e) = 2\alpha \Re B(Q_1, Q_2) + 2\alpha \tilde{B}(Q_1, Q_2, k_e).
\]

The explicit formulae for the above \( \Re B \) and \( \tilde{B} \) are given in Ref. [3]; we do not repeat them here because they are rather lengthy. This simplification corresponds to neglecting the ISR–WSR interference terms in the YFS form factor.

3. \( \tilde{S} \) factors:
In these we also neglect the ISR–WSR interferences:

\[
\tilde{S}(p_1, p_2, Q_1, Q_2, k) \rightarrow \tilde{S}_I(p_1, p_2, k) + \tilde{S}_W(Q_1, Q_2, k),
\]

and compensate this by the weight

\[
w_{\tilde{S}} = \prod_{i=1}^n \frac{\tilde{S}(p_1, p_2, Q_1, Q_2, k)}{\tilde{S}_I(p_1, p_2, k) + \tilde{S}_W(Q_1, Q_2, k)},
\]

where \( \tilde{S}_I \) and \( \tilde{S}_W \) are given in Eq. (8).

After these simplifications, we obtain from Eq. (4) the formula for the first-level, “crude” cross section:

\[
\sigma_{\text{Crude}} = \sum_{n=0}^\infty \frac{1}{n!} \int d\sigma_1 d\sigma_2 \frac{d^3 Q_1}{Q_1^0} \frac{d^3 Q_2}{Q_2^0} \prod_{i=1}^4 \frac{d^3 q_i}{q_i^0} \left[ \prod_{i=1}^n \frac{d^3 k_i}{k_i^0} \right] \delta(\epsilon' - \epsilon) \theta(k_i^0 - k_e) \prod_{i=1}^n \left\{ \tilde{S}_I(k_i) + \tilde{S}_W(k_i) \right\} e^{Y_I + Y_W} \sum_{\varphi \in \{I, W\}^n} p_\varphi b_0 (\{p, Q, q\} R_\psi).
\]
The above distribution can be generated using the MC algorithms of YFS3 and KoralW. To make this more transparent, we shall write this formula in a slightly different but equivalent form.

The product of the $\tilde{S}$-factors in the square brackets of Eq. (43) together with the sum over the photon partition can be written as

$$
\prod_{i=1}^{n} \frac{d^{3}k_{i}}{k_{i}^{0}} \left\{ \tilde{S}_{I}(k_{i}) + \tilde{S}_{W}(k_{i}) \right\} \theta(k_{i}^{0} - k_{\epsilon}) \sum_{\varphi \in \{I, W\}^{n}} p_{\varphi} b_{0} \left\{ \{p, Q, q\}^{R_{\varphi}} \right\} =
$$

$$
\sum_{\varphi \in \{I, W\}^{n}} \prod_{\varphi_{i}=I}^{n} \frac{d^{3}k_{i}}{k_{i}^{0}} \tilde{S}_{I}(k_{i}) \theta(k_{i}^{0} - k_{\epsilon}) \prod_{\varphi_{j}=W}^{n} \frac{d^{3}k_{j}}{k_{j}^{0}} \tilde{S}_{W}(k_{j}) \theta(k_{j}^{0} - k_{\epsilon}) b_{0} \left\{ \{p, Q, q\}^{R_{\varphi}} \right\},
$$

where we have substituted Eq. (7) for the partition weight $p_{\varphi}$.

After some algebra, and exploiting the Bose–Einstein symmetry for photons, the formula of Eq. (43) can be written as

$$
\sigma_{\text{Crude}} = \sum_{n_{I}=0}^{\infty} \sum_{n_{W}=0}^{\infty} \int ds' \frac{d^{3}P'}{P'^{0}} \delta^{(4)} \left( p_{1} + p_{2} - \sum_{i=1}^{n_{I}} k_{i} - P' \right) \int ds'_{1} \frac{d^{3}Q_{1}}{Q_{1}^{0}} \frac{d^{3}Q_{2}}{Q_{2}^{0}}
$$

$$
\left[ \frac{1}{n_{I}} \prod_{i=1}^{n_{I}} \frac{d^{3}k_{i}}{k_{i}^{0}} \tilde{S}_{I}(k_{i}) \theta(k_{i}^{0} - k_{\epsilon}) \right] \left[ \frac{1}{n_{W}} \prod_{j=1}^{n_{W}} \frac{d^{3}k_{j}}{k_{j}^{0}} \tilde{S}_{W}(k_{j}) \theta(k_{j}^{0} - k_{\epsilon}) \right]
$$

$$
\delta^{(4)} \left( P' - Q_{1} - Q_{2} - \sum_{j=1}^{n_{W}} k_{j} \right) \prod_{i=1}^{4} \frac{d^{3}q_{i}}{q_{i}^{0}} \delta^{(4)} \left( Q_{1} - q_{1} - q_{2} \right) \delta^{(4)} \left( Q_{2} - q_{3} - q_{4} \right)
$$

$$
e^{Y_{I}} e^{Y_{W}} b_{0} \left\{ \{p, Q, q\}^{R_{\varphi}} \right\},
$$

where $s' = P'^{2}$ is the CMS energy squared in the ISR-reduced (effective) frame, and $n_{I}$ and $n_{W}$ are the multiplicities of the ISR and WSR photons, respectively.

To generate the ISR and WSR according to Eq. (45), we can use the MC algorithm of YFS3 (see Refs. [47, 48]) for multiphoton radiation in the fermion-pair production, where the initial-final-state interferences were neglected. The only modification we had to do in this algorithm was to extend it to the case of heavy particles of unequal masses (see Ref. [48]). Generating particular values of $n_{I}$ and $n_{W}$ corresponds to choosing a given photon partition for $n = n_{I} + n_{W}$ in the sum of Eq. (44). This means that the sum over partitions is “randomized” in the program, i.e. not evaluated directly but generated using the MC methods. For the given (generated) photon partition, we then calculate the weights $w_{\varphi}^{\beta}$, $w_{Y}$ and $w_{S}$. Such a solution can be regarded as a multibranch MC algorithm, see Ref. [50], where each branch corresponds to a particular photon partition.
3.2 MC Weights and Absolute Normalization

In the process of simplifying the fully differential distribution of Eq. (4)

\[ \sigma_{\text{Best}} = \sum_{n=0}^{\infty} \frac{1}{n!} \int \text{dLips}_n(p, q, k_1, \ldots, k_n) \rho_{n, \text{Best}}^n(p, q, k_1, \ldots, k_n) \]  

(46)
described in the previous section we have finished with the “crude differential distribution” \( \rho_{\text{Crude}} \) inside the integral

\[ \sigma_{\text{Crude}} = \sum_{n_I=0}^{\infty} \sum_{n_W=0}^{\infty} \frac{1}{n_I! n_W!} \int \text{dLips}_n(p, q, k_1, \ldots, k_n) \rho_{n_I, n_W}^{\text{Crude}}(p, q, k_1, \ldots, k_n), \]  

(47)
where \( n = n_I + n_W \). Using the above notation, the total correcting weight is defined as follows:

\[ w^{(1)} = \frac{\rho_{n, \text{Best}}^n(p, q, k_1, \ldots, k_n)}{\rho_{n_I, n_W}^{\text{Crude}}(p, q, k_1, \ldots, k_n)} = w_\beta^0 w_Y w_\tilde{S}. \]  

(48)

In the actual program it comes in several versions, because for the purpose of the technical tests, and for the discussion of the physical precision, we need to switch off/on certain contributions in the full distribution \( \rho_{\text{Best}}^n \).

However, in the actual MC program we do not actually generate the distribution \( \rho_{\text{Crude}} \), but a certain, more primitive, “primary” distribution \( \rho_{\text{Prim}} \). For its definition and full description, and the description of how events are generated according to this distribution, starting from the uniform random numbers, we refer the reader to the most complete documentation of the YFS3 Monte Carlo algorithm in Ref. [48]; Refs. [21] and [34] can also be helpful. The weight correcting for the transition from the primary to crude distribution is given by

\[ w^{(2)} = \frac{\rho_{n_I, n_W}^{\text{Crude}}(p, q, k_1, \ldots, k_n)}{\rho_{n_I, n_W}^{\text{Prim}}(p, q, k_1, \ldots, k_n)} \]  

(49)
and is fully defined in Ref. [48]. The total MC weight is of course equal to the product

\[ w = w^{(1)} w^{(2)} \]  

(50)
or

\[ w = w^\text{Model} w^\text{Crude}, \]  

(51)
with

\[ w^\text{Model} = w_\beta^0 b_0, \quad w^\text{Crude} = w^{(2)} w_Y w_\tilde{S} b_0^{-1}, \]  

(52)
where \( w^\text{Model} \) is called in the program the “model weight” (this weight and its many variants are provided in the WtSet array; see Section 5), and \( w^\text{Crude} \) is called the “crude weight” (in the program it is equal to the product WtCrud1*WtCrud2; see Section 5).
The integrated cross section is given by the product of the average weight and the integrated primary cross section

\[
\sigma = \langle w \rangle \sigma_{\text{Prim}} = \langle w \rangle \int d\sigma_{\text{Prim}}.
\]  

(53)

The integrated primary cross section we also call a “normalization cross section”, because it provides the absolute normalization for the whole MC calculation.

### 4 Structure of the Program

In this section we provide the reader with a brief guide of the YFSWW3 program. We shall describe its main routines, libraries and interfaces. We want to note here that the program, in the distribution version, is prepared for a UNIX/Linux-type operating system that supports directories and the make utility. The source code is distributed over a number of subdirectories, in order to make the structure of the program more transparent and easier to handle. Also a system of the Makefile’s is provided, for compilation, execution and other auxiliary functions (e.g. clean-up). (Of course, this organization, which is in fact rather simple, can be avoided and the whole source code can be put into a single FORTRAN file.) In the following we give a review of all YFSWW3 subdirectories. We start from the directories that are most interesting from the user’s point of view, then going to the ones with more technical contents. In the distribution package, all these subdirectories are located in the main YFSWW3 directory: yfsww3-1.16-export. This directory also comprises two important files: README and RELEASE.NOTES, which we recommend the user look through before using the program. They contain some basic information about YFSWW3: how to compile/link and run the program, a brief documentation of the code, etc.

#### 4.1 demo – Demonstration Program

The subdirectory demo contains a demonstration program in the file demo.f. Generally, the user is supposed to provide his/her own main program; nevertheless, the file quoted here provides a simple example of such a program. It has a double role: (1) as a useful template, and (2) as a first cross-check that the MC generator YFSWW3 runs correctly on a given installation. The essential part of this program is a loop in which a series of MC events is generated. It also reads the input from a disk file, but no histogramming is performed and most of the output comes from the generator itself. At the end of the program, a MC-integrated cross section of YFSWW3 is compared with the Born-level and ISR results from the semi-analytical program KorWan. The program is compiled/link and executed, with the help of Makefile, for the input data set given in the files demo.input. The program also reads the default settings provided in the file data_DEFAULTS located in the directory data_files – some of these default settings are overridden by data from demo.input. The demonstration program is run for unweighted
events with the external libraries PHOTOS, TAUOLA and JETSET switched on. The output, written into the disk file demo.output, can then be compared with the one provided in the file demo.output.linux. It was obtained on a PC Intel Pentium III under the Linux RedHat 6.1 operating system.

4.2 data_files – Default Input Data

The subdirectory data_files contains the input data file data_DEFAULTS, which gives the defaults settings for YFSWW3. It is identical with the one used in the program KoralW, so the two programs can be set-up from the same input data file! This considerably facilitates a combination the results of the two programs; see Ref. [25] for more details. Some entries in this data file are dummy for YFSWW3 and are kept only for compatibility with KoralW, while some of them are specific to YFSWW3 and are treated as dummy parameters in KoralW. They are accompanied by appropriate comments. This file can also be used as a template to create the user’s own input data file, where only those entries that will have different settings from the default data file may be included. The user input file should be read after the file data_DEFAULTS, so that the user settings can override the default ones; see the example in the demo.f file described in the previous subsection. This directory comprises also some older data files kept for backward compatibility. Their names contain, after a dot, the last two digits of the year of their creation. (The latest of these files is identical with the file data_DEFAULTS.)

4.3 yfsww – Master Unit

The subdirectory yfsww contains the actual Monte Carlo event generator. Subprograms that can be used directly are the following:

• YFSWW_ReadDataX – the subprogram used to read, from the disk file, the default input data of YFSWW3 and subsequently the data of the user into the array xpar at the very beginning of the use of YFSWW3.

• YFSWW_Initialize – the subprogram that does all initializations of internal variables. First, it sets-up the main parameters of the program (through the routine filexp) according to the values in the input data files, then it calls several initializers of the main internal modules of the generator, such as KarLud(-1,...), and of external packages, such as TAUOLA, etc. It prints out directly or indirectly all the input parameters.

• YFSWW_Make – the most important subprogram of YFSWW3. It generates single MC events. Functionally, it is a high-level management subprogram in the event generation. It invokes (through the routine yfsww3) other routines that perform specific tasks, such as: generation of ISR photons (KarLud), generation of radiative photons from W’s (KarFin), generation of the W’s and final 4 f four-momenta (WW_Presam), evaluation of the Coulomb correction (WTCoul), calculation of the matrix element
(Model), of the YFS form factor, of the interference corrections, etc. It builds up the total MC weight from the weights supplied by these routines. This weight is then returned, in the weighted-event mode, as the main event weight, or used in the rejection loop (in the routine yfsww3), in the unweighted-event mode, for constructing the weight = 1 event. It also calls PHOTOS, which generates photon radiation in the $W$ decays, TAUOLA for decaying of the final-state $\tau$’s (if they appear), and JETSET to perform fragmentation/hadronization of the final-state quarks. It keeps track, through special monitoring routines, of all the information necessary for the final results from the MC event generation.

- **YFSWW_Finalize** does all final bookkeeping, including the calculation of the integrated (total) cross section. It prints a summary output for the whole sample of generated MC events.

The first subprogram is located in the file `readata.f`, while the following three are in the file `yfsww3.f`.

This directory contains also a collection of utility routines, which can be used, for instance, to access information on particle flavours and four-momenta, on event weights, etc. They are located in the file `ww_get.f` (see the comments in this file for more details).

### 4.4 model – Matrix Elements

The routines for the “model” weights calculations are located in the subdirectory `model`.

The master routine here is the subroutine `Model` (in the file `model.f`); by calling some other routines, this evaluates the “model” weights $w_{\text{Model}}$ as defined in Eq. (52).

1. For the **Born-level matrix element** we use the same routines as in the program KoralW – they are collected in the file `born.f`. The same again are used for the ISR corrections, located in the file `betas.f`.

2. The $O(\alpha)$ **virtual and real soft photon corrections** are evaluated with the help of a special interface – the routine `VirSof` – in the file `virsof.f`. In the first call, this routine makes all the necessary initializations, and then calculates the EW corrections by calling appropriate routines from the EW library for each generated event. The pretabulated (or “fast”) version of the EW correction is invoked through the routine `DnlFast` (in the file `dnlfast.f`). This routine makes a pretabulation of the EW corrections in the initialization stage of the program and then, during the event generation, it performs the linear interpolation of the pretabulated values to calculate the EW corrections for each event as described in Section 2.2.2.

3. The $O(\alpha)$ **hard photon matrix element** in the $WW$ production, which contributes to the function $\beta_1$, see Eq. (21), is calculated with the help of the routine `eewwg`, located in the file `eewwg.f`. This routines evaluates appropriate spin amplitudes for the process $e^+e^- \rightarrow W^+W^-\gamma$ according to the formulas of Ref. [8], which are then combined coherently with the spin amplitudes for $W$ decays.
In order to calculate all these contributions to the $\beta$ functions, the subroutine `Model` calls the routines that perform appropriate “reduction” procedures as described in Section 2. Except for calculating the contributions to the YFS exponentiated cross section, this routine provides also the weights corresponding to the strict $\mathcal{O}(\alpha^0)$ and $\mathcal{O}(\alpha^1)$ calculations. In the end, the subroutine `Model` fills the array of weights with the weights corresponding to: (i) several variants of the SM models (for example different orders of QED ISR) and (ii) various components in the differential cross section for a given variant of the SM (for example contributions from various $\beta$’s). The subroutine `Model` returns the best “model” weight.

The routine `WTCoul` – in the file `coulco.f` – for the Coulomb (both “standard” and “screened”) correction calculation is also located in this directory.

4.5 **ewc** – Electroweak Library

The subdirectory `ewc` contains the routines for the calculation of the electroweak (virtual plus soft photon) corrections in the on shell $WW$ production process according to Refs. [7, 11]. For the evaluation of one-loop integrals they use the package `FF` of Ref. [51], which is located in a separate directory (see below).

4.6 **interfs** – Interfaces to External Libraries

Interfaces to some external libraries are collected in the directory `interfs`. The most important routines are:

- `inietc` – in the file `tauola_photos_ini.f` – sets up some parameters of `PHOTOS` and `TAUOLA`.
- `tohep` – in the file `hepface.f` – fills in the `/HEPEVT/ COMMON` block, calls `PHOTOS` for photon radiation in $W$ decays and `TAUOLA` for $\tau$ decays.
- `tohad` – in the file `lundface.f` – calls `JETSET` for the quark fragmentation/hadronization.

Note that the `COMMON` block `/HEPEVT/` is expected to contain single-precision (`REAL*4`) variables.

4.7 **wdeclib** – External Libraries for $W$ Decay

The subdirectory `wdeclib` provides the following programs:

- `PHOTOS` [27] for photon radiation in the $W$ decays (up to two photons),
- `TAUOLA` [52] for $\tau$ decays with radiative corrections,
- `JETSET` version 7.4 [53] for quark fragmentation/hadronization,
located in the files: photos.f, tauola.f and jetset74.f, respectively.

In YFSWW3 1.16 we use the recently modified PHOTOS, where also the photon radiation from quarks can be activated with the help of a special input parameter switch; see the tables with the input parameters in Appendix A. Note that in the distribution version of TAUOLA the parameters in the \(\tau\)-decay modes are not adjusted to the recent experimental data. We recommend the user to replace this version of TAUOLA with the one accepted within his/her own collaboration. The technical update described in Refs. [4, 5] will resolve this inconvenience in the future releases of the program.

4.8 glib – Histogramming and Numerical Library

A handy FORTRAN histogramming package, GlibK [56], is provided in the subdirectory glib (the file glibk.f). It is used by YFSWW3 both for hard-coded internal bookkeeping and for some optional “external” tests. The package is similar in its usage to the classic HBOOK of CERNLIB. This subdirectory also contains the useful numerical package yfslib.f with the routines for weight monitoring, random-number generation, numerical integration, one- and two-dimensional adaptive MC sampling, Lorentz transformations, etc.

4.9 semian – Semi-Analytical Program KorWan

The subdirectory semian contains the package KorWan for the semi-analytical calculations of the cross section for the off-shell \(WW\) production and decay at the Born level and with the ISR corrections (using the structure-function formalism). Also the Coulomb effect is included in KorWan in both versions: the “standard” and the “screened” ones. The package is described in detail in Refs. [21–23]. It can be used for a quick consistency check of the YFSWW3 results.

4.10 ff – FF Package

In the subdirectory ff is located the FF package [51] for the evaluation of one-loop integrals needed to calculate the \(O(\alpha)\) EW virtual corrections.

4.11 dok – Related Papers

In the subdirectory dok, we include some of our papers related to YFSWW3. They are in the form of gzip’ed PostScript files.

4.12 rewtt – Reweighting Tools

The subdirectory rewtt contains routines that are not needed for the standard MC event generation but for various event reweighting purposes. We provide here the tools for two kinds of reweighting: the reweighting (1) of the KoralW generated events to correct
for $O(\alpha)$ NL effects in $WW$ production, and (2) of the YFSWW3 generated events to take into account effects due to some input parameters (typically $W$-mass) changes. All these tools are described in detail in the following; see also the file README provided in this subdirectory.

### 4.12.1 Reweighting KoralW Events by YFSWW3

The main tool for this kind of reweighting is the function $\text{YFSWW}_\text{WtNL}$, located in the file `rewt_K.f`. For the events, in terms of flavours and four-momenta of the final-state fermions and the ISR photons, provided through its parameters, this function calculates and returns the corresponding correction weight from YFSWW3. This weight is a ratio of the fully differential distribution. In practice, we calculate the ratio of the MC weights corresponding to two distributions – this is more convenient. We can do it because the common crude distribution entering both weights cancels out, for a given number of real photons of Eq. (5), to the reference distribution in the “ISR approximation” defined in Eq. (30), which is the same as in KoralW in the CC03 mode (Eq. (4) of Ref. [23]). It is defined in Eq. (33). In the reweighting mode of YFSWW3, we implement an approximate version of this weight:

$$w_{NL} \equiv \frac{\rho_{n}^{\text{Best}}(\{p, Q, q\}^{R_I}, k_1, \ldots, k_n)}{\rho_{n}^{R}(\{p, Q, q\}^{R_I}, k_1, \ldots, k_n)} = 1 + \delta_{NL-}, \quad (54)$$

where $\rho_{n}^{\text{Best}}$ is a variant of $\rho_{n}^{\text{Best}}$ in which the sum over the photon partitions is reduced to only one term, in which all photons are associated to the ISR with the partition probability $p_{\rho} = 1$, and $R_I$ means the respective reduction procedure. Unfortunately, we loose in this way a (small) part of the genuine $O(\alpha)$ corrections. The full $O(\alpha)$ correction can be included only if YFSWW3 is run in the event-generation mode, and the correction weight for reweighting (accounting for the $4f$ background) is supplied from KoralW. (This leads, however, to large MC weight fluctuations, see below.) We have checked, however, that the bias introduced in the LEP2 energy range by this approximation is within 0.1%, both for the total cross section and for distributions. This can be explained by the fact that the $W$’s, as heavy and slowly moving particles, do not radiate much at these energies. Of course, the main weight of YFSWW3 based on $\rho_{n}^{\text{Best}}$ does not have this bias. In this way, by assigning this correction weight to the KoralW events, one can include, in a simple way, the $O(\alpha)$ NL correction $\delta_{NL-}$ into the KoralW results. How to do it in practice, we explain in detail in Ref. [25].

### 4.12.2 Reweighting YFSWW3 Events by YFSWW3

This kind of reweighting can be useful in some data analyzes, e.g. when one has accumulated a large sample of simulated events for a given input parameter set-up and one wants to assess the effects of changing some of the input parameters, but without having to repeat the full simulation with the new input parameters. The tools for such a reweighting are collected in the file `rewt_Y.f`. We provide two methods: “exact” and
“approximate”. In the exact method the partition vector for photons is stored along with their four-momenta, while in the approximate method the partition information is not recorded.

(a) The exact method:
The tool for the exact method is SUBROUTINE YFSWW\_WME(WtME), where WtME corresponds to the weight \( w^{\text{Model}} \) of Eq. (52). It requires storing on the disk/tape the entire content of four COMMON blocks: /MOMINI/, /MOMFIN/, /MOMDEC/ and /DeChan/ during the event-generation run of YFSWW3. We assume that the appropriate tools for writing the above information on the disk/tape as well as for reading it in the reweighting run of the program are provided by the user of the code (see the source code for more details).

How to use it:

1. The primary-run generating/storing events:
   - Run YFSWW3 and store on the disk/tape the whole content of the COMMON blocks: /MOMINI/, /MOMFIN/, /MOMDEC/, /DeChan/. This is done for some input parameters set-up (A).

2. The secondary-run reweighting events:
   - Initialize the program with the input parameters set-up (A) used for the event generation.
   - Instead of generating events, call YFSWW\_WME(WtME\_A) for all stored events and save the computed matrix element weight WtME\_A (in some arbitrary units).
   - Initialize the program with a different input parameters set-up (B).
   - Calculate the weight WtME\_B and save as above.
   - For each event, calculate the ratio \( Wt = \frac{WtME\_B}{WtME\_A} \) and use it as the weight for reweighting the original sample of events.
   - This procedure can be repeated for any new input parameters set-up (B).

Example of running the program in the reweighting mode:

```fortran
CALL YFSWW\_ReaDataX(’./data\_DEFAULTS’,1,10000,xpar) ! reading general defaults
CALL YFSWW\_ReaDataX(’./user\_input’ ,0,10000,xpar) ! reading user input
CALL YFSWW\_Initialize(xpar) ! initialize generator
NevTot = 1000 ! number of events to be reweighted
DO Iev = 1, NevTot
   CALL YFSWW\_WME(WtME) ! calculate matrix element weight for stored events
ENDDO
```

For more information about running YFSWW3, see Section 5.

(b) The approximate (simpler) version:

In the case when one does not have all the information needed for the exact (recommended) method of reweighting (because of the lack of disk/tape space or some other reasons), one can use
\text{WtME} = \text{YFSWW\_WME\_Smpl(Iflav, pf1, pf2, pf3, pf4, Phot, Nphot)}

instead of \text{CALL YFSWW\_WME(WtME)}. In this case one needs to store for each event only
the flavours and four-momenta of the final-state fermions and the number and the four-
momenta of all photons. \textbf{Important}: They should come from \textit{YFSWW3} itself, not from
\textit{PHOTOS} after the FSR. These four-momenta and flavours are available from two \texttt{COMMON}
blocks:

\begin{verbatim}
    COMMON / DECAYS / Iflav(4), amdec(4)
    COMMON / MOMDEC / pf1(4), pf2(4), pf3(4), pf4(4), Phot(100, 4), Nphot
\end{verbatim}

(the fermion mass array \texttt{amdec(4)} does not need to be stored). They can be alternatively
accessed with the help of two getter-routines:

\begin{verbatim}
    SUBROUTINE YFSWW\_Get4f(flav, p1, p2, p3, p4)
    SUBROUTINE YFSWW\_GetPhotAll(NphAll, PhoAll)
\end{verbatim}

located in the file \texttt{ww\_get.f} in the subdirectory \texttt{yfsww}. The first one provides the flavours
and four-momenta of the final-state 4-fermions, while the second one provides the number
and four-momenta of the radiative photons (excluding the ones from \texttt{PHOTOS}).

The \texttt{FUNCTION YFSWW\_WME\_Smpl} fills the appropriate \texttt{COMMON} blocks and calculates (returns) the matrix element weight (corresponding to \texttt{WtME} in the previous method). This
method is \textit{approximate} because it does not use the information on the photon partitions
but assigns all the photons to the ISR in performing the partition-dependent reduction
procedure (described in Section 2.2). Our tests show, however, that the differences be-
tween this and the exact method are numerically negligible, e.g. \(< 0.01\% for the total
cross section at \(\sqrt{s} = 200 \text{ GeV.}\)

\subsection{Reweighting \texttt{YFSWW3} Events by \texttt{KoralW}}

Similar reweighting tools are provided in the new version of \texttt{KoralW} \cite{25}. They can be used
for reweighting events generated by \texttt{YFSWW3} with the weight provided by \texttt{KoralW} in order
to correct for the missing 4\textit{f}-background contribution to the signal \(WW\) process. The
relevant reweighting routines in \texttt{KoralW} require as an input the events from \texttt{YFSWW3}, which
are generated according to the ISR-type reference differential distribution. Such events are
provided by \texttt{YFSWW3} in the standard generation mode, with the help of the getter-routine
\texttt{YFSWW\_GetEvtISR}, located in the file \texttt{ww\_get.f} of the directory \texttt{yfsww}. The correction
weight provided by \texttt{KoralW} can be used to reweight the constant-weight (unweighted)
events from \texttt{YFSWW3} or to correct the \texttt{YFSWW3} weights in the variable-weight (weighted-
event) mode; see Ref. \cite{25} for more details.

\section{How to use the Program}

In this section we provide a short guide of how to use the current version of \texttt{YFSWW3}, and
we describe the input parameters of the program and its output.
5.1 Principal Entries of YFSWW3

The principal entries of the YFSWW3 package, which the user has to call in his/her application in order to generate a series of MC events, were already listed and described briefly in Subsection 4.3. Here we shall add more information on their functionality. The calling sequence constituting a typical Monte Carlo run will look as follows:

```
CALL YFSWW_ReadDataX('./data_DEFAULTS',1,10000,xpar) ! reading general defaults
CALL YFSWW_ReadDataX('./user.input' ,0,10000,xpar) ! reading user input
CALL YFSWW_Initialize(xpar) ! initialize generator
DO loop = 1, 10000 ! loop over MC events
    CALL YFSWW_Make ! generate single MC event
ENDDO
CALL YFSWW_Finalize ! final bookkeeping, print
CALL YFSWW_GetXSecMC(XSecMC,XErrMC) ! get total cross section
```

In the first call of YFSWW_ReadDataX, default data are read into the REAL*8 array xpar(10000). The YFSWW3 has almost no data hidden in the source code (this is not true for TAUOLA and JETSET). The file data_DEFAULTS, which is read first, is placed in the distribution subdirectory data_files. This file provides necessary initial default values of all input parameters. The user should never modify it. It can be copied to a local directory or, better, a symbolic link should be created to the original data_files/data_DEFAULTS. This file is quite sizeable and the user is usually interested only in changing some subset of these data. In the second call of YFSWW_ReadDataX, the user can overwrite the default data with his/her own smaller set of input data, which are placed in the user.input file. For example, the simplest input data, defining only the CMS energy, would look like this:

```
BeginX
*<ia><----data-----><------------------comments------------->
  1  190d0 CMSEne =CMS total energy [GeV]
EndX
```

As we see, data cards begin with the keyword BeginX and end with the keyword EndX. The comment lines are allowed – they begin with * in the first column. The data themselves are in a fixed format, with the index i of the array xpar(i) followed by the data value and a trailing comment. The example of the input data set for the demonstration program demo.f in the subdirectory demo provides a useful template for typical user’s data. The complete set of data in data_DEFAULTS is described in detail in Tables 1–5, see Appendix A. Obviously, the user is interested in manipulating only some of them and will retain the default values in most of the cases.

The YFSWW.Initialize is invoked to initialize the generator. It reads the input data from the array xpar, prints them out and sends them down to the various modules and auxiliary libraries. The programs have to be called strictly in the same order as in the above example. At this point one is ready to generate the series of MC events. The generation of a single event is done with the help of YFSWW.Make. After the generation loop is completed, we may invoke YFSWW_Finalize, which does the final bookkeeping,
prints out various pieces of information on the MC run, and calculates the total MC integrated cross section (in pb). In order to obtain this cross section the user may call the routine \texttt{YFSWW\_GetXSecMC(XSecMC,XErrMC)}.

### 5.2 Input/Output Parameters

As we explained in the previous section, the input parameters enter through the \texttt{xpar} array, being a parameter of the routine \texttt{YFSWW\_Initialize}. Their meaning is explained in Tables 1–5 of Appendix A.

The principal output of \texttt{YFSWW3} is the Monte Carlo event, which is just a list of final-state four-momenta (in GeV) and flavours, encoded in the standard \texttt{/HEPEVT/} event record. In the present version, we still provide the \texttt{REAL*4} version of \texttt{/HEPEVT/} of dimension 2000. If the user is interested in the parton momenta before hadronization, then, in addition to \texttt{/HEPEVT/}, they are available (see also Table 8) through

\begin{verbatim}
CALL YFSWW\_Get4f(p1,p2,p3,p4) ! get final 4f flavours and 4-momenta
CALL YFSWW\_GetBeams(q1,q2)   ! get beam 4-momenta
CALL YFSWW\_GetPhotAll(NphAll,PhoAll) ! get photon multiplicity and 4-momenta
\end{verbatim}

Alternatively, all the four-momenta are available from the internal \texttt{COMMON} blocks \texttt{/MOMWWP/} and \texttt{/MOMDEC/}, see Tables 6 and 7 for details.

In the case of a MC run with weighted events, the user is provided with the main weight (see also Table 10) through

\begin{verbatim}
CALL YFSWW\_GetWtMain(WtMain)     ! get main Monte Carlo weight
\end{verbatim}

Of course, for unweighted events \texttt{WtMain=1}. For special purposes the user may also be interested in auxiliary weights, which are provided (see also Table 11) with the help of

\begin{verbatim}
CALL YFSWW\_GetWtAll(WtMain,WtCrud,WtSetAll) ! get all Monte Carlo weights
\end{verbatim}

where \texttt{REAL*8 WtSetAll(100)} is an array of the weights described in Table 9. The definition of the main weight \texttt{WtMain} depends on the input switches \texttt{KeyCor} and \texttt{KeyLPA}, see Table 9; for instance, the best model for \texttt{KeyCor = 5} and \texttt{KeyLPA = 0} corresponding to Eq. (4) is defined as follows:

\begin{verbatim}
WtMod = Wtcrui*Wtcrui2*(WtSetAll(41)-WtSetAll(2)+WtSetAll(4)).
\end{verbatim}

Alternatively, the auxiliary weights \texttt{WtSet} defined in Eq. (52) are available from the \texttt{COMMON /WGTALL/}, see Table 11. The total auxiliary weight should be defined as:

\begin{verbatim}
WtAuxEW(i) = WtCru1*WtCru2*WtSetAll(i)
\end{verbatim}

for \(O(\alpha)\) EW corrections, and

\begin{verbatim}
WtAuxISR(i) = WtCru1*WtCru2*WtSetAll(i)
\end{verbatim}

for the ISR corrections. The corresponding integrated cross section is simply obtained by multiplying the average of the total weight by the \texttt{normalization} cross section of Eq. (53) (or the “primary cross section” in the terminology of Ref. [48]), which is provided through
CALL YFSWW_GetXSecNR(XSecNR,XErrNR) ! get normalization x-section

Note that we expect the user to exploit the auxiliary weights only in the variable-weight operation mode. It is, however, not impossible to use them also for the constant-weight events. In such a case we recommend the user to contact the authors for more instructions on how to do it correctly.

The complete description of the post-generation output parameters from YFSWW_Finalize is collected in Table 11.

5.3 Printouts of the Program

In this section we describe a printout of the demonstration program demo in the “best” mode, shown in Appendix B. The printout starts with the detailed specification of the actually used input parameters. Also, logos of some of the activated subprograms (KarLud, KarFin) and libraries (PHOTOS, TAUOLA, JETSET) are printed here. Next, the printout of one full event (in the standard PDG convention) is shown (in the actual demo.output file, five events can be seen). Then, the printouts from the post-generation mode, i.e. CALL YFSWW_Finalize, appear. First, one can see summary reports from the subprograms KarLud and KarFin. Their meaning is rather technical, so the average user of YFSWW3 does not have to worry about them (unless something unusual appears there). One thing that the user may check from time to time there is the entry B5 in the KarFin window B: the average value of the WCTRL weight should be equal to 1 within the statistical error (if it is not, please notify the authors!). After these technical printouts from the YFSWW3 subprograms, there are several summary reports from TAUOLA on the \( \tau \) decays. The final reports of the main YFSWW3 MC generator are collected in three windows: A, B and C.

The window A contains the most important information from the user’s point of view. It provides the values of:

A0 : the CMS energy in GeV;

A1 : the best-order total cross section with its absolute statistical error for the generated statistics sample (in pb);

A2 : the relative error of the above cross section;

A3 : the total number of generated events;

A4 : the number of accepted events;

A5 : the number of events with negative weights;

A6 : the relative contribution to the total cross section from the events with the negative weights;

A7 : the number of overweighted events;

A8 : the relative contribution to the total cross section from the over-weighted events;
A9 : the value of the maximum weight for event rejection;

A10 : the average weight.

The numbers in the entries A5, A6, A7, A8 should be as small as possible. In particular, the relative contributions to the cross section from the negative weights and the overweights should be much smaller than the expected accuracy of the MC calculations. If there is a large contribution from the overweighted events, one should try to increase the value of the maximum weight \( WtMax \), see Table 1. In the case of a large negative weight contribution, one should contact the authors. The ratio of the average weight (entry A10) to the maximum weight (entry A9) shows the efficiency of the MC algorithm in the unweighed events generation, i.e. the event acceptance rate. In the example given in the demo program, it is \( \sim 20\% \).

The window B is devoted to the technical information on the ISR corrections in different orders in \( \alpha \) within the YFS framework. This information is quite important, because it shows how big the contributions of subsequent orders of the perturbative series are, and thus allows us to estimate the missing higher-order effects. The entries B1–B4 provide the values of the total cross section (in pb) in the orders \( \mathcal{O}(\alpha^0)_{\text{exp}} - \mathcal{O}(\alpha^3)_{\text{exp}} \) of the ISR, while the entries B15–B17 give the differences of these values in subsequent orders. The entries B5–B14 contain the information on the contributions from the individual YFS ISR residuals \( \bar{\beta}_{i,\text{ISR}} \), also in different orders in \( \alpha \). The differences between various residuals and various orders for a given residual are provided in the entries B18–B26.

The window C contains the information on the exact \( \mathcal{O}(\alpha^1)_{\text{exp}} \) corrections to the WW production stage. The first entry (C1) shows the total \( \mathcal{O}(\alpha^1)_{\text{exp}} \) cross section for the exact EW corrections calculation, the second one (C2) shows the same for the approximate (“fast”) EW corrections, and the third one (C3) contains the \( \mathcal{O}(\alpha^1)_{\text{exp}} \) LL cross section. The second part of this window shows the differences between the exact and approximate EW corrections (C4), and between the exact and the LL results (C5). The final part (entries C6–C8) contains the differences of various \( \bar{\beta}_i \) contributions. All these values are given in pb.

This completes the description of the output of YFSWW3. The remaining entries shown in the demo.output file (e.g. histograms) are produced by the demo main program.

6 Summary

We presented the Monte Carlo event generator YFSWW3 version 1.16 for the combined \( W \)-pair production and decay process. It includes the complete \( \mathcal{O}(\alpha) \) EW corrections in the WW production process in the leading-pole approximation, in addition to other numerically sizeable but physically less interesting effects, such as QED ISR effects up to \( \mathcal{O}(\alpha^3) \) in the LL approximation. The program is precise enough to obtain a Standard Model prediction for the total cross section and any distribution at LEP2, understanding that the contribution from the background diagrams is included in the calculation with the help of KoralW (or removed from the data). YFSWW3 includes programming tools to communicate
with KoralW in the process of generating a single MC event. The YFSWW3 program was also tested and is applicable for the energy range up to 1.5 TeV (the LC/TESLA range). The next possible improvement in the program is the introduction of the complete YFS exponentiation in the MC simulation of the W decays.

Acknowledgments

We acknowledge the support of the CERN TH and EP Divisions, all the LEP Collaborations and the DESY Directorate. We would like to thank all members of the LEP2 WW/4f Working Group for many useful discussions, particularly R. Chierici, M. Grünewald, A. Valassi and M. Verzocchi for valuable feedback concerning the development of the code, and the authors of RACOONWW for the useful numerical comparisons.
A Program Parameters and Their Settings

| Parameter  | Position and meaning                                                                 |
|------------|--------------------------------------------------------------------------------------|
| CMSEne     | xpar(1) (=200.0): $\sqrt{s}$, centre-of-mass (CMS) energy [GeV]                      |
| Gmu        | xpar(2) (=1.16639d-5): $G_F$, Fermi constant [GeV$^{-2}$]                             |
| alfWin     | xpar(3) (=128.07d0): $1/\alpha_W$ inverse QED coupling constant at $M_W$ scale       |
| aMaZ       | xpar(4) (=91.1882): $M_Z$, mass of $Z$ boson, [GeV]                                   |
| GammZ      | xpar(5) (=2.4952): $\Gamma_Z$, width of $Z$ boson [GeV]                              |
| aMaW       | xpar(6) (=80.419): $M_W$, mass of $W$ boson [GeV]                                     |
| GammW      | xpar(7) (=2.120): $\Gamma_W$, width of $W$ boson [GeV], for ($\text{gammw} < 0$) $\Gamma_W$ is recalculated from $G_\mu$, $M_W$ and $\alpha_S$ |
| VVmin      | xpar(8) (=1d-6): Minimum $v$-variable (dimensionless), IR cut-off                      |
| VVmax      | xpar(9) (=0.99): Maximum value of $v$-variable                                        |
| WtMax      | xpar(10) (=2.0): Maximum weight for rejection, for $wtmax < 0$ redefined inside the program |
| aMH        | xpar(11) (=115.0): Higgs mass [GeV]                                                   |
| aGH        | xpar(12) (=1.0): Higgs width [GeV]                                                    |
| alpha_s    | xpar(13) (=0.1185): QCD coupling constant                                             |
|            | xpar(14-19): Dummy parameters in YFSWW3 (specific to KoralW)                          |

Table 1: The list of input parameters of the YFSWW3 generator in the xpar vector. The default values are in brackets.

| Parameter  | Position and meaning                                                                 |
|------------|--------------------------------------------------------------------------------------|
| KeyCor     | xpar(2001) (=5): Radiative corrections switch                                       |
|            | =0: Born                                                                            |
|            | =1: Above + ISR                                                                     |
|            | =2: Above + Coulomb correction                                                      |
|            | =3: Above + Full YFS form factor for $WW$ production                                |
|            | =4: Above + Radiation from $WW$                                                    |
|            | =5: Above + Exact $O(\alpha)$ EW corrections in $WW$ production (BEST!)            |
|            | =6: Same as $5$ but with pretabulated EWC (approximate but faster!)                |
| KeyLPA     | xpar(2002) (=0): LPA mode switch                                                    |
|            | =0/1: LPA$_a$/LPA$_b$ (LPA$_a$ recommended, LPA$_b$ for test only)               |

Table 2: The list of input parameters in the xpar vector specific to the YFSWW3 generator (not used in KoralW). The default values are in brackets.
| Parameter   | Position and meaning                                                                                                                                                                                                 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $g_1(1)$    | $=$DCMPLX(xpar(21),xpar(31)) = $g_1^z$, for $WWZ$ vertex                                                                                                                                                           |
| $\kappa(1)$| $=$DCMPLX(xpar(22),xpar(32)) = $\kappa_z$, for $WWZ$ vertex                                                                                                                                                       |
| $\lambda(1)$| $=$DCMPLX(xpar(23),xpar(33)) = $\lambda_z$, for $WWZ$ vertex                                                                                                                                                     |
| $g_4(1)$    | $=$DCMPLX(xpar(24),xpar(34)) = $g_{4z}$, for $WWZ$ vertex                                                                                                                                                         |
| $g_5(1)$    | $=$DCMPLX(xpar(25),xpar(35)) = $g_{5z}$, for $WWZ$ vertex                                                                                                                                                         |
| $\kappa_{pt}(1)$| $=$DCMPLX(xpar(26),xpar(36)) = $\tilde{\kappa}_z$, for $WWZ$ vertex                                                                                                                                           |
| $\lambda_{pt}(1)$| $=$DCMPLX(xpar(27),xpar(37)) = $\tilde{\lambda}_z$, for $WWZ$ vertex                                                                                                                                       |
| $g_1(2)$    | $=$DCMPLX(xpar(41),xpar(51)) = $g_{1g}$, for $WW\gamma$ vertex                                                                                                                                                  |
| $\kappa(2)$| $=$DCMPLX(xpar(42),xpar(52)) = $\kappa_g$, for $WW\gamma$ vertex                                                                                                                                               |
| $\lambda(2)$| $=$DCMPLX(xpar(43),xpar(53)) = $\lambda_g$, for $WW\gamma$ vertex                                                                                                                                              |
| $g_4(2)$    | $=$DCMPLX(xpar(44),xpar(54)) = $g_{4g}$, for $WW\gamma$ vertex                                                                                                                                                  |
| $g_5(2)$    | $=$DCMPLX(xpar(45),xpar(55)) = $g_{5g}$, for $WW\gamma$ vertex                                                                                                                                                  |
| $\kappa_{pt}(2)$| $=$DCMPLX(xpar(46),xpar(56)) = $\tilde{\kappa}_g$, for $WW\gamma$ vertex                                                                                                                                       |
| $\lambda_{pt}(2)$| $=$DCMPLX(xpar(47),xpar(57)) = $\tilde{\lambda}_g$, for $WW\gamma$ vertex                                                                                                                                     |
| $\delta Z$ | =xpar(61) = $\delta_Z$                                                                                                                                                                                          |
| $x_\gamma$  | =xpar(62) = $x_\gamma$                                                                                                                                                                                           |
| $x_Z$       | =xpar(63) = $x_Z$                                                                                                                                                                                                |
| $y_\gamma$  | =xpar(64) = $y_\gamma$                                                                                                                                                                                           |
| $y_Z$       | =xpar(65) = $y_Z$                                                                                                                                                                                                |
| $\alpha_{W\phi}$| =xpar(71) = $\alpha_{W\phi}$                                                                                                                                                                                   |
| $\alpha_{B\phi}$| =xpar(72) = $\alpha_{B\phi}$                                                                                                                                                                                   |
| $\alpha_W$  | =xpar(73) = $\alpha_W$                                                                                                                                                                                           |
| $\alpha_{mel}$| =xpar(100) (=0.510998902d-3): beam (electron) mass                                                                                                                                                                 |
| $AlfInvl$   | =xpar(101) (=137.03599976): $1/\alpha_{QED}$ at the Thomson limit                                                                                                                                                |
| $gpicob$    | =xpar(101) (=389.379292d6): GeV$^{-2}$ to picobarn translation constant                                                                                                                                            |
| BR(1:20)    | xpar(131-139): $W$ branching ratios (BR); the numbering of entries is: 1 = $ud$, 2 = $cd$, 3 = $us$, 4 = $cs$, 5 = $ub$, 6 = $cb$, 7 = $e\nu_e$, 8 = $\mu\nu_\mu$, 9 = $\tau\nu_\tau$ |
| amafin(20)  | xpar(500 +10*KF +6): Masses of the $W$ decay products; the used entries KF (in PDG notation) are: 1 = $d$, 2 = $u$, 3 = $s$, 4 = $c$, 5 = $b$, 6 = $t$, 11 = $e$, 12 = $\nu_e$, 13 = $\mu$, 14 = $\nu_\mu$, 15 = $\tau$, 16 = $\nu_\tau$. |
| VCKM(1:3,1:3)| xpar(111-119): CKM matrix elements (PDG 2000)                                                                                                                                                                      |

Table 3: The list of input parameters of the YFSWW3 generator in the xpar vector (cont.). The default values are in brackets.
| Parameter | Position and meaning |
|-----------|----------------------|
| KeyCul xpar(1011-1013): | Dummy parameters in YFSWW3 (specific to KoralW) |
| xpar(1014) (=2): | Coulomb correction switch |
| =0: | Coulomb correction is OFF |
| =1: | “Standard” Coulomb correction is ON |
| =2: | “Screened” Coulomb correction is ON |
| KeyBra xpar(1021) (=2): | Sets W branching ratios (BR), used for normalization of the matrix element |
| =0: | Born values (no mixing) with “naive” QCD (if \( \alpha_S = 0 \): \( ud, cs = 1/3, e\nu, \mu\nu, \tau\nu = 1/9, \text{ others} = 0 \)) |
| =1: | Values of BRs taken from the input |
| =2: | With CKM mixing and “naive” QCD, calculated in the IBA from the CKM matrix (PDG 2000) |
| KeyMas xpar(1022): | Dummy parameter in YFSWW3 (specific to KoralW) |
| KeyZet xpar(1023) (=1): | Z width choice |
| =0: | Z width in Z propagator: \( (s/M_Z)\Gamma_Z \) |
| =1: | Z width in Z propagator: \( M_Z\Gamma_Z \) |
| =2: | no Z width in Z propagator |
| KeySpn xpar(1024) (=1): | Spin effects in W decays |
| =0: | Spin effects OFF (for tests only) |
| =1: | Spin effects ON (recommended) |
| KeyRed xpar(1025): | Dummy parameter in YFSWW3 (specific to KoralW) |
| KeyWu xpar(1026) (=1): | W width choice |
| =0: | W width in W propagator: \( (s/M_W)\Gamma_W \) |
| =1: | W width in W propagator: \( M_W\Gamma_W \) |
| =2: | No W width (\( \Gamma_W = 0 \)) in W propagator (dangerous!) |
| KeyWgt xpar(1031) (=0): | Unweighted/Weighted choice |
| =0: | Unweighted events WtMod=1, for apparatus Monte Carlo |
| =1: | Weighted events (variable weights) |
| KeyRnd xpar(1032) (=1): | Random number generator switch |
| =1: | RANMAR random number generator |
| =2: | ECURAN random number generator |
| =3: | CARRAN random number generator |
| KeySmp xpar(1033): | Dummy parameter in YFSWW3 (specific to KoralW) |
| KeyMix xpar(1041) (=1): | Input parameter scheme (IPS) switch |
| =0: | LEP2 ’95 Workshop scheme (only for Born and ISR) |
| =1: | \( G_\mu \)-scheme (recommended; the only choice for \( O(\alpha) \) EW corrections) |
| =2: | \( \alpha \)-scheme (“pure” Born; for tests) |
| Key4f xpar(1042): | Dummy parameter in YFSWW3 (specific to KoralW) |

Table 4: The list of input parameters of the YFSWW3 generator in the xpar vector (cont.). The default values are in brackets.
| Parameter | Position and meaning |
|-----------|----------------------|
| KeyAcc    | xpar(1043) (=0): Anomalous TGCs switch  
=0: Anomalous WWV couplings OFF  
>0: Anomalous WWV couplings ON  
=1: The most general (complex number) TGCs in the notation of Hagiwara et al., Nucl. Phys. B282 (1987) 253  
=2 Parametrization of CERN 96-01, Vol. 1, p. 525: $\delta_Z, x_\gamma, x_Z, y_\gamma, y_Z$  
=3: Parametrization of CERN 96-01, Vol. 1, p. 525: $\alpha_W, \alpha_B, \alpha_W$ |
| KeyZon    | xpar(1044): Dummy parameter in YFSWW3 (specific to KoralW) |
| KeyWon    | xpar(1045): Dummy parameter in YFSWW3 (specific to KoralW) |
| KeyDwm    | xpar(1055) (=0): Sets decay channel of $W^-$  
=0: Inclusive; otherwise the exclusive modes for the $W$ are:  
\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
ud & cd & us & cs & ub & cb & ev & \mu\nu & \tau\nu
\end{bmatrix}
\] |
| KeyDwp    | xpar(1056) (=0): Sets the decay channel of $W^+$, similar assignments as for $W^-$ |
| Nout      | xpar(1057) (=0): Output unit number for the generator (if < 0 then Nout=16) |
| Jak1      | xpar(1071) (=0): Input for TAUOLA, defines decay mode of the $\tau^+$ in $W^+$ decay |
| Jak2      | xpar(1072) (=0): Input for TAUOLA, defines decay mode of the $\tau^-$ in $W^-$ decay |
| Jak1,Jak2 | Jak1,Jak2 = -1: TAUOLA is switched OFF  
Jak1,Jak2 = 0: Requests all $\tau^\pm$ decay channels to be simulated  
Jak1,Jak2 > 0: Single specific $\tau^\pm$ decay channel, see TAUOLA manual |
| Itdkrc    | xpar(1073) (=1): Input for TAUOLA, radiative corrections in leptonic $\tau$ decays switch  
Itdkrc = 1: Corrections are ON  
Itdkrc = 0: Corrections are OFF |
| IfPhot    | xpar(1074) (=1): PHOTOS activation switch  
=0: PHOTOS is OFF  
=1: Radiation in leptonic $W$ decays is ON  
=2: Radiation in both leptonic and quarkonic $W$ decays is ON (for tests) |
| IfHadM    | xpar(1075) (=1): $W^-$ hadronization activation switch (JETSET)  
IfHadM,ifhadP=0: Hadronization is OFF  
IfHadM,ifhadP=1: Hadronization is ON |
| IfHadP    | xpar(1075) (=1): $W^+$ hadronization activation switch (JETSET)  
In the present version IfHadM and IfHadP have to be equal! |
| Umask     | xpar(1101-1302): Dummy parameters in YFSWW3 (specific to KoralW) |
| BE params | xpar(4061-4070): Dummy parameters in YFSWW3 (specific to KoralW) |

Table 5: The list of input parameters of the YFSWW3 generator in the xpar vector (cont.). The default values are in brackets.
| Parameter       | Meaning                                                                 |
|-----------------|-------------------------------------------------------------------------|
| Q1(4)           | Four-momentum of $W^-$                                                  |
| Q2(4)           | Four-momentum of $W^+$                                                  |
| SPhum(4)        | Sum of four-momenta of photons from the $WW$ production stage           |
| SPhot(100,4)    | Four-momenta of photons from the $WW$ production stage                  |
| Nphot           | Number of photons from the $WW$ production stage                        |

Table 6: The list of four-momenta in the COMMON /MOMWWP/ of the YFSWW3 generator. They are given in GeV in the CMS of the incoming beams.

| Parameter       | Meaning                                                                 |
|-----------------|-------------------------------------------------------------------------|
| pf1(4)          | Four-momentum of the fermion from the $W^-$ decay                       |
| pf2(4)          | Four-momentum of the antifermion from the $W^-$ decay                   |
| pf3(4)          | Four-momentum of the fermion from the $W^+$ decay                       |
| pf4(4)          | Four-momentum of the antifermion from the $W^+$ decay                   |
| Phot(100,4)     | Four-momenta of photons from the $WW$ production stage                  |
| Npho            | Multiplicity of photons from the $WW$ production stage                  |

Table 7: The list of four-momenta in the COMMON /MOMDEC/ of the YFSWW3 generator. They are given in GeV in the CMS of the incoming beams.

| Routine         | Parameter       | Meaning                                                                 |
|-----------------|-----------------|-------------------------------------------------------------------------|
| YFSWW_GetBeams  | q1(4)           | Four-momentum of the $e^-$ beam                                          |
|                 | q2(4)           | Four-momentum of the $e^+$ beam                                          |
| YFSWW_Get4f     | flav(4)         | Flavours of the final 4f-state in the PDG convention                     |
|                 | p1(4)           | Four-momentum of the fermion of flav(1)                                  |
|                 | p2(4)           | Four-momentum of the fermion of flav(2)                                  |
|                 | p3(4)           | Four-momentum of the fermion of flav(3)                                  |
|                 | p4(4)           | Four-momentum of the fermion of flav(4)                                  |
| YFSWW_GetPhotAll| NphAll          | Number of photons from the $WW$ production stage                        |
|                 | PhoAll(100,4)   | Four-momenta of these photons                                           |

Table 8: The list of output parameters of the YFSWW3 generator transferred through parameters of the getter-routines: YFSWW_GetBeams(q1,q2), YFSWW_Get4f(flav,p1,p2,p3,p4) and YFSWW_GetPhotAll(NphAll,PhoAll).
Parameter | Meaning
--- | ---
WtCru1 | Crude weight for ISR
WtCru2 | Crude weight for the rest
WtMod | They are necessary to build the total weight out of WtSet
       | The “best” weight corresponding to Eq. (4), e.g. for KeyCor=5:
       | $WtMod = WtCru1 \times WtCru2 \times (WtSet(41) - WtSet(2) + WtSet(4))$
WtSet(1-19) | Various ISR-type contributions to the matrix element ($\bar{\beta}$ functions) – to get the total weight they must be multiplied by WtCru1
WtSet(1) | Zeroth-order ($\bar{\beta}_0$)
WtSet(2) | First-order ISR ($\bar{\beta}_0 + \bar{\beta}_1$)
WtSet(3) | Second-order ISR ($\bar{\beta}_0 + \bar{\beta}_1 + \bar{\beta}_2$)
WtSet(4) | Third-order ISR ($\bar{\beta}_0 + \bar{\beta}_1 + \bar{\beta}_2 + \bar{\beta}_3$)
WtSet(10) | $O(\alpha^0)$ contribution from $\bar{\beta}_0$
WtSet(11) | $O(\alpha^1)$ ISR contribution from $\bar{\beta}_0$
WtSet(12) | $O(\alpha^1)$ ISR contribution from $\bar{\beta}_1$
WtSet(13) | $O(\alpha^2)$ ISR contribution from $\bar{\beta}_0$
WtSet(14) | $O(\alpha^2)$ ISR contribution from $\bar{\beta}_1$
WtSet(15) | $O(\alpha^2)$ ISR contribution from $\bar{\beta}_2$
WtSet(16) | $O(\alpha^3)$ ISR contribution from $\bar{\beta}_0$
WtSet(17) | $O(\alpha^3)$ ISR contribution from $\bar{\beta}_1$
WtSet(18) | $O(\alpha^3)$ ISR contribution from $\bar{\beta}_2$
WtSet(19) | $O(\alpha^3)$ ISR contribution from $\bar{\beta}_3$
WtSet(41-52) | Contributions to the matrix element from $O(\alpha)$ EW corrections in the $WW$ production stage – to get the total weight they must be multiplied by WtCru1\*WtCru2
WtSet(41) | $O(\alpha^1)$ EW corrections ($\bar{\beta}_0 + \bar{\beta}_1$)
WtSet(51) | Pretabulated (“fast”) $O(\alpha^1)$ EW corrections ($\bar{\beta}_0 + \bar{\beta}_1$)
WtSet(42) | $O(\alpha^1)$ contribution from $\bar{\beta}_0$
WtSet(43) | $O(\alpha^1)$ contribution from $\bar{\beta}_1$
WtSet(52) | Pretabulated $O(\alpha^1)$ contribution from $\bar{\beta}_0$
WtSet(65-68) | Fixed-order (no exponentiation) contributions (for tests)
WtSet(65) | $O(\alpha^0)$, no exponentiation
WtSet(66) | $O(\alpha^1)$, no exponentiation
WtSet(67) | $O(\alpha^1)$ virtual + soft real photon only, no exponentiation
WtSet(68) | $O(\alpha^1)$ real hard photon only, no exponentiation

Table 9: The list of output weights in the COMMON /WGTALL/ of the YFSWW3 generator.
### Table 10: The list of output parameters of the YFSWW3 generator transferred through parameters of the getter-routines: YFSWW_GetWtMain(WtMain), YFSWW_GetWtISR(WtISR) and YFSWW_GetWtAll(WtMain,WtCrud,WtSetAll).

| Routine       | Parameter | Meaning                                           |
|---------------|-----------|---------------------------------------------------|
| YFSWW_GetWtMain | WtMain    | *Principal* best MC event weight                   |
| YFSWW_GetWtISR | WtISR     | Best ISR-type weight                               |
| YFSWW_GetWtAll | WtMain    | *Principal* best MC event weight                   |
|               | WtCrud    | Crude MC event weight: WtCrud=WtCru1*WtCru2        |
|               | WtSetAll(100) | Model weights (the same as in WtSet(100) in COMMON /WGTALL/) |

### Table 11: The list of output parameters of the YFSWW3 generator transferred through parameters of the getter-routines: YFSWW_GetXSecMC(XSecMC,XErrMC), YFSWW_GetXSecNR(XSecNR,XErrNR) and YFSWW_GetNevMC(NevMC).

| Routine      | Parameter | Meaning                                           |
|--------------|-----------|---------------------------------------------------|
| YFSWW_GetXSecMC | XSecMC    | *Principal* total MC cross section [pb]            |
|              | XErrMC    | Its absolute error [pb]                           |
| YFSWW_GetXSecNR | XSecNR    | Normalization (primary) cross section, see Sect. 5.2: |
|              | XErrNR    | For KeyWgt=0, the principal cross section XSecMC [pb], |
|              |           | For KeyWgt=1, the crude cross section XCrude [pb]  |
|              |           | Its absolute error [pb]                           |
| YFSWW_GetNevMC | NevMC     | Total number of generated Monte Carlo events      |
B Output of the Demo Program

****************************************************************************

YFSWW3 version 1.16, January 2001
S. Jadach, W. Placzek, M. Skrzypek
B.F.L. Ward, Z. Was

INPUT PARAMATERS:

200.00000000 CMS energy total CMSEne I.0

5 Rad. Corr. switch KeyCor IR1

2101 QED super-switch KeyRad IQ1

1 Init. state Rad. KeyISR IQ2

0 Final state Rad. KeyFSR IQ3

1 Next-To-Leading KeyNLL IQ4

2 Coulomb corr. KeyCul IQ5

101112 Physics super-switch KeyPhy IP1

0 FS mass reduction KeyRed IP2

1 Spin in W decays KeySpn IP3

1 Z proparg. KeyZet IP4

1 Mass kinematics. KeyMas IP5

2 Branching Rat. KeyBra IP6

1 W proparg. KeyWu IP7

10 Technical super-swic

0 LPA_a or LPA_b mode KeyLPA IT2

0 Presampler type KeySmp IT3

1 Rand. Numb. Gen. 1 KeyRnd IT4

0 Weighting switch KeyWgt IT5

10001 Miscellaneous KeyMis IM1

1 EW Input Par. Schem KeyMix IM2

0 4 fermion matr el Key4f IM3

0 Anomalous couplings KeyAcc IM4

1 WW type final state KeyWon IM5

0 ZZ type final state KeyZon IM6

0 W- decay mode KeyDwm ID1

0 W+ decay mode KeyDwp ID2

1.16639000 G_mu * 1d5 Gmu I.1

132.50494581 Inverse alpha_w alfWin I.2

91.18820000 Z mass [GeV] aMaZ I.3

2.49520000 W mass [GeV] aMaW I.5

80.41900000 W width [GeV] GammW I.6

0.000000100 Dummy infrared cut VVmin I.7

0.990000000 v_max ( =1 ) VMax I.8

4.00000000 Max wt for rejectn. WtMax I.9

4.00000000 Max wt for CC03 rej WtMax I.10

0.11850000 alpha_s: QCD coupl. alphas I11

0.000000000 Color Re-Con. Prob. PReco I12

0.22224994 sin(theta_W)**2 sinW2 I13

Z width in Z propagator: M_Z *GAMM_Z

CKM matrix elements:

0.97493000 V_ud VCKM(1,1) IV1

0.22250000 V_us VCKM(1,2) IV2

0.00350000 V_ub VCKM(1,3) IV3

-0.22246000 V_cd VCKM(2,1) IV4

44
* 0.97412000 \( V_{cs} \) VCKM(2,2) IV5 *
* 0.04000000 \( V_{cb} \) VCKM(2,3) IV6 *
* 0.00549000 \( V_{td} \) VCKM(3,1) IV7 *
* -0.03978000 \( V_{ts} \) VCKM(3,2) IV8 *
* 0.99920000 \( V_{tb} \) VCKM(3,3) IV9 *

**Unitarity check of the CKM matrix:**

\[
V^+ V = \begin{pmatrix}
1.0000 & 0.0000 & 0.0000 \\
0.0000 & 1.0000 & 0.0000 \\
0.0000 & 0.0000 & 1.0000
\end{pmatrix}
\]

**Branching ratios of \( W \) decays:**

\[
\begin{align*}
ud & \text{ BR}(1) IB1 \\
\text{cd} & \text{ BR}(2) IB2 \\
\text{us} & \text{ BR}(3) IB3 \\
\text{cs} & \text{ BR}(4) IB4 \\
\text{ub} & \text{ BR}(5) IB5 \\
\text{cb} & \text{ BR}(6) IB6 \\
\text{e} & \text{ BR}(7) IB7 \\
\text{mu} & \text{ BR}(8) IB8 \\
\text{tau} & \text{ BR}(9) IB9
\end{align*}
\]

**Fermion masses:**

\[
\begin{align*}
d & \text{ amafin}(1) IM1 \\
u & \text{ amafin}(2) IM2 \\
s & \text{ amafin}(3) IM3 \\
c & \text{ amafin}(4) IM4 \\
b & \text{ amafin}(5) IM5 \\
t & \text{ amafin}(6) IM6 \\
e & \text{ amafin}(11) IM7 \\
ve & \text{ amafin}(12) IM8 \\
mu & \text{ amafin}(13) IM9 \\
vmu & \text{ amafin}(14) IM10 \\
\text{tau} & \text{ amafin}(15) IM11 \\
\text{vtau} & \text{ amafin}(16) IM12
\end{align*}
\]

**Higgs mass:**

\[115.00000000 \text{ Higgs aMHig IMH}\]

**DECAY LIBRARIES**

\[
\begin{align*}
0 & \text{ TAUOLA for } W^+ \text{ Jak1 IL1} \\
0 & \text{ TAUOLA for } W^- \text{ Jak2 IL2} \\
1 & \text{ TAUOLA Ord}(\alpha) \text{ Itdkrc IL3} \\
1 & \text{ PHOTOS IfPhot IL4} \\
1 & \text{ JETSET for } W^- \text{ IfHadM IL5} \\
1 & \text{ JETSET for } W^+ \text{ IfHadP IL6}
\end{align*}
\]
| Index | Particle | Isospin | Charge | Mass (MeV) | Width (MeV) | Lifetime (ns) | Cross Section (pb) | Branching Ratio (%) |
|-------|----------|---------|--------|------------|-------------|---------------|-------------------|---------------------|
| 1     | e^-      | 1       | 0      | 0.000      | 0.000       | 159.64        | 0.000             | 100.000             |
| 2     | e^+      | 1       | 0      | 0.000      | 0.000       | 159.64        | 0.000             | 100.000             |
| 3     | mu^-     | 1       | 0      | 0.000      | 0.000       | 105.66        | 0.000             | 100.000             |
| 4     | mu^+     | 1       | 0      | 0.000      | 0.000       | 105.66        | 0.000             | 100.000             |
| 5     | pi^-     | 0       | 0      | 0.000      | 0.000       | 135.00        | 0.000             | 100.000             |
| 6     | pi^+     | 0       | 0      | 0.000      | 0.000       | 135.00        | 0.000             | 100.000             |

**Event listing (standard)**

| Event | Particle | Isospin | Charge | Mass (MeV) | Width (MeV) | Lifetime (ns) | Cross Section (pb) | Branching Ratio (%) |
|-------|----------|---------|--------|------------|-------------|---------------|-------------------|---------------------|
| 1     | e^-      | 1       | 0      | 0.000      | 0.000       | 159.64        | 0.000             | 100.000             |
| 2     | e^+      | 1       | 0      | 0.000      | 0.000       | 159.64        | 0.000             | 100.000             |
| 3     | mu^-     | 1       | 0      | 0.000      | 0.000       | 105.66        | 0.000             | 100.000             |
| 4     | mu^+     | 1       | 0      | 0.000      | 0.000       | 105.66        | 0.000             | 100.000             |
| 5     | pi^-     | 0       | 0      | 0.000      | 0.000       | 135.00        | 0.000             | 100.000             |
| 6     | pi^+     | 0       | 0      | 0.000      | 0.000       | 135.00        | 0.000             | 100.000             |
235 gamma 1 22 187 0 0 0.08510 -0.01401 -0.06558 0.10835 0.000 00
236 gamma 1 22 187 0 0 2.98739 -0.10576 -3.21645 4.39104 0.000 00
238 p+ 1 22 12 188 0 0 2.76880 -0.33055 -3.14146 4.30403 0.9383 0
239 pi- 1 -21 12 188 0 0 0.65387 0.03010 -0.65573 0.93698 0.1396 0
240 pi+ 1 21 12 188 0 0 0.26095 0.08270 -0.24512 0.39307 0.1396 0
241 p~- 1 -22 12 239 0 0 3.53794 0.36931 -4.30069 5.65949 0.938 30
242 pi+ 1 21 12 239 0 0 0.44102 0.13979 -0.50348 0.69787 0.1396 0

sum charge: 0.00 sum momentum and inv. mass: -0.00001

*************************************************************************** KarLud FINAL REPORT *
* window A ** 4537 total no of events NEVTOT A0 *
* 0 WT<0 events NEVNEG A1 *
* 19.33542764 xs_cru VESKO [pb] XCVESK A2 *
* 19.17361574 ++ 0.01754039 xs_est VESKO [pb] XSVE A3 *
* 19.19161478 ++ 0.00001919 xs_est Gauss [pb] XSGS A4 *
* -0.00093786 ++ 0.00091582 XCVE/XCGS-1 A5 *
* 251.14598768 ++ 0.24820021 <WTKLAR> WTKARL A6 *
***************************************************************************

*************************************************************************** KarFin output - window A *
* Weight Statistics *
* 0.99643450 +- 0.00368336 general weight WT A1 *
* 4263 generated events NEVGEN A2 *
* 0.08139808 aver. ph. multi. AVMULT A3 *
* 0 Marked photons MARTOT A4 *
* 19.1080609 ++ 0.00145252 Kinematics, smin WT1 A5 *
* 0.9999743 ++ 0.00000168 Jacobian WT2 A6 *
* 1.00534843 ++ 0.00335486 Photon ang. dist. WT3 A7 *
***************************************************************************

*************************************************************************** KarFin output - window B *
* on mass weights *
* 0.96568167 +- 0.00245351 removal wgt WTREM WT6 B1 *
* 4263 no of raw events B2 *
* 0.08139808 raw ph. multipl. B4 *
* 0.99951442 +- 0.00245437 control wgt WCTRL WT5 B5 *
* 0.10000000E-05 epsilon B6 *
* 0.10000000E-08 delta B7 *
***************************************************************************

*************************************************************************** ******** DADMEL FINAL REPORT ******** *
* 48 NEVRAW = NO. OF EL DECAYS TOTAL *
* 18 NEVACC = NO. OF EL DECS ACCEPTED *
* 0 NEVOVR = NO. OF OVERWEIGHTED EVENTS *
* 0.48927E-12 PARTIAL WTDTH ( ELECTRON) IN GEV UNITS *
* 1.207993507 IN UNITS GFERMI**2*MASS**5/192/PI**3 *
* 0.128391966 RELATIVE ERROR OF PARTIAL WIDTH *
* COMPLETE QED CORRECTIONS INCLUDED *
* BUT ONLY V-A CUPLINGS *
***************************************************************************

****skipped****
### YFSWW3 Final Report

**Window A**

| CMS energy total | CMSENE | A0      |
|------------------|--------|---------|
| 200.00000000     |        |         |
| 16.9645667       | 0.24052939 | XSMCPB | A1 |
| 0.01417843       | relative error | EREL | A2 |
| 0.00000000       | -- 0.00000000 | XSNEG | A6 |
| 17.20746824      | 0.21080089 | xsec total | 0(alf0) | B1 |
| 17.22391080      | 0.21107044 | xsec total | 0(alf2) | B3 |
| 17.22440793      | 0.21107682 | xsec total | 0(alf3) | B4 |
| 16.71805108      | 0.20216719 | xsec total | 0(alf6) | B6 |
| 17.0699748       | 0.21412627 | xsec(beta00) | 0(alf0) | B6 |
| 0.48533564       | -- 0.1785827 | xsec(total) | 0(alf2) | B9 |
| 0.00022486       | 0.00031913 | xsec(beta20) | 0(alf2) | B10 |
| 0.170752722      | -- 0.00002720 | xsec(beta03) | 0(alf3) | B14 |
| 0.48941716       | 0.01874270 | xnalph | 0(alf1) | B15 |

**Window B**

| CMS energy total | CMSENE | A0      |
|------------------|--------|---------|
| 16.71805108      | 0.20216719 | xsec total | 0(alf0) | B1 |
| 17.0746824       | 0.21080089 | xsec total | 0(alf1) | B2 |
| 17.22391080      | 0.21107044 | xsec total | 0(alf2) | B3 |
| 17.22440793      | 0.21107205 | xsec total | 0(alf3) | B4 |
| 16.71805108      | 0.20216719 | xsec(beta00) | 0(alf0) | B6 |
| 17.0699748       | 0.21412627 | xsec(beta02) | 0(alf2) | B8 |
| 0.48533564       | -- 0.1785827 | xsec(beta11) | 0(alf2) | B9 |
| 0.00022486       | 0.00031913 | xsec(beta20) | 0(alf2) | B10 |
| 0.170752722      | -- 0.00002720 | xsec(beta03) | 0(alf3) | B14 |
| 0.48941716       | 0.01874270 | xsec total | 0(alf1) | B15 |

**xsec_tot differences**

| CMS energy total | CMSENE | A0      |
|------------------|--------|---------|
| 0.00000000       | -- 0.01243983 | xsec total | 0(alf1) | B15 |
**YFSWW3 final report**

* Window C: ** O(\(\alpha_f\)) EWRC in WW-prod. *

* 16.94721804 ++ 0.24020843 \(x_{\text{tot}}: \text{EW-ex}\) O(\(\alpha_f\)) C1 *
* 16.92711416 ++ 0.24008944 \(x_{\text{tot}}: \text{EW-ap}\) O(\(\alpha_f\)) C2 *
* 17.20746824 ++ 0.21080089 \(x_{\text{tot}}: \text{ISR-LL}\) O(\(\alpha_f\)) C3 *
* \(x_{\text{sec-tot}}\) differences *
* 0.02010388 ++ 0.00091149 \(\text{EW-ex} - \text{EW-ap}\) O(\(\alpha_f\)) C4 *
* -0.26025020 +++-0.11286284 \(\text{EW-ex} - \text{ISR-LL}\) O(\(\alpha_f\)) C5 *
* \(\text{betas differences}\) *
* 0.02010388 ++ 0.00091149 \(\text{bet01(ex - ap)}\) O(\(\alpha_f\)) C6 *
* -0.40008677 +++-0.00639368 \(\text{bet01(ex - LL)}\) O(\(\alpha_f\)) C7 *
* -0.02807084 +++-0.00454222 \(\text{bet10(ex - LL)}\) O(\(\alpha_f\)) C8 *

**window C: O(\(\alpha_f\)) EWRC in WW-prod.**

YFSWW3 final report

0(\(\alpha_f\)) EWRC in WW-prod.

**Window C:**

\[
\begin{align*}
0(\alpha_f) & = 16.94721804 \pm 0.24020843 \\
0(\alpha_f) & = 16.92711416 \pm 0.24008944 \\
0(\alpha_f) & = 17.20746824 \pm 0.21080089
\end{align*}
\]

**xsec tot differences**

\[
\begin{align*}
0(\alpha_f) & = 0.02010388 \pm 0.00091149 \\
0(\alpha_f) & = -0.26025020 \pm 0.11286284
\end{align*}
\]

**betas differences**

\[
\begin{align*}
0(\alpha_f) & = 0.02010388 \pm 0.00091149 \\
0(\alpha_f) & = -0.40008677 \pm 0.00639368 \\
0(\alpha_f) & = -0.02807084 \pm 0.00454222
\end{align*}
\]

..... skipped ....

**window C: O(\(\alpha_f\)) EWRC in WW-prod.**

YFSWW3 final report

0(\(\alpha_f\)) EWRC in WW-prod.

**Window C:**

\[
\begin{align*}
0(\alpha_f) & = 16.94721804 \pm 0.24020843 \\
0(\alpha_f) & = 16.92711416 \pm 0.24008944 \\
0(\alpha_f) & = 17.20746824 \pm 0.21080089
\end{align*}
\]

**xsec tot differences**

\[
\begin{align*}
0(\alpha_f) & = 0.02010388 \pm 0.00091149 \\
0(\alpha_f) & = -0.26025020 \pm 0.11286284
\end{align*}
\]

**betas differences**

\[
\begin{align*}
0(\alpha_f) & = 0.02010388 \pm 0.00091149 \\
0(\alpha_f) & = -0.40008677 \pm 0.00639368 \\
0(\alpha_f) & = -0.02807084 \pm 0.00454222
\end{align*}
\]

..... skipped ....
References

[1] *Physics at LEP2*, edited by G. Altarelli, T. Sjöstrand, and F. Zwirner (CERN 96-01, Geneva, 1996), 2 vols.

[2] *Reports of the Working Groups on Precision Calculations for LEP2 Physics*, edited by S. Jadach, G. Passarino, and R. Pittau (CERN 2000-009, Geneva, 2000).

[3] S. Jadach, W. Placzek, M. Skrzypek, and B. F. L. Ward, Phys. Rev. D54, 5434 (1996).

[4] S. Jadach *et al.*, Phys. Lett. B417, 326 (1998).

[5] S. Jadach *et al.*, Phys. Rev. D61, 113010 (2000).

[6] S. Jadach *et al.*, Precision Predictions for (Un)Stable $W^+W^-$ Production At and Beyond LEP2 Energies, preprint CERN-TH/2000-337; hep-ph/0007013; to be submitted to Phys. Lett. B.

[7] J. Fleischer, F. Jegerlehner, and M. Zralek, Z. Phys. C42, 409 (1989).

[8] K. Kołodziej and M. Zralek, Phys. Rev. D43, 3619 (1991).

[9] J. Fleischer, K. Kołodziej, and F. Jegerlehner, Phys. Rev. D47, 830 (1993).

[10] J. Fleischer, K. Kołodziej, and F. Jegerlehner, Phys. Rev. D49, 2174 (1994).

[11] J. Fleischer, F. Jegerlehner, K. Kołodziej, and G. J. van Oldenborgh, Comput. Phys. Commun. 85, 29 (1995).

[12] A. Denner, S. Dittmaier, M. Roth, and D. Wackeroth, Phys. Lett. B475, 127 (2000).

[13] A. Denner, S. Dittmaier, M. Roth, and D. Wackeroth, Nucl. Phys. B587, 67 (2000).

[14] W. Beenakker, F. A. Berends, and A. Chapovsky, Nucl. Phys. B548, 3 (1999).

[15] W. Beenakker *et al.*, WW Cross-Sections and Distributions, in Ref. [1], Vol. 1, p. 79.

[16] A. Vicini, Acta Phys. Polon. B29, 2847 (1998).

[17] D. Bardin *et al.*, Event Generators for WW Physics, in Ref. [1], Vol. 2, p. 3.

[18] M. Grünewald *et al.*, Four-Fermion Production in Electron-Positron Collisions, in Ref. [2], p. 1.

[19] R. G. Stuart, Nucl. Phys. B498, 28 (1997), and references therein.

[20] W. Placzek *et al.*, W-Pair Production with YFSWW/KoralW, preprint CERN-TH/2000-357; hep-ph/0012094.
[21] M. Skrzypek, S. Jadach, W. Placzek, and Z. Wąs, Comput. Phys. Commun. 94, 216 (1996).

[22] M. Skrzypek et al., Phys. Lett. B372, 289 (1996).

[23] S. Jadach et al., Comput. Phys. Commun. 119, 272 (1999).

[24] M. Skrzypek and Z. Wąs, Comput. Phys. Commun. 125, 8 (2000).

[25] S. Jadach et al., The Monte Carlo Program KoralW version 1.51 and The Concurrent Monte Carlo KoralW*YFSWW3 with All Background Graphs and First Order Corrections to W-pair Production, preprint CERN-TH/2001-040, UTHEP-01-0102, February 2001, (to be submitted to Comput. Phys. Commun.).

[26] S. Jadach et al., in preparation.

[27] E. Barberio, B. van Eijk, and Z. Wąs, Comput. Phys. Commun. 66, 115 (1991), ibid. 79 291 (1994).

[28] V. S. Fadin, V. A. Khoze, A. Martin, and W. Stirling, Phys. Lett. B363, 112 (1995).

[29] A. P. Chapovsky and V. A. Khoze, Eur. Phys. J. C9, 449 (1999).

[30] W. Beenakker, A. P. Chapovsky, and F. A. Berends, Nucl. Phys. B508, 17 (1997).

[31] W. Beenakker, A. P. Chapovsky, and F. A. Berends, Phys. Lett. B411, 203 (1997).

[32] A. Denner, S. Dittmaier, and M. Roth, Nucl. Phys. B519, 39 (1998).

[33] A. Denner, S. Dittmaier, and M. Roth, Phys. Lett. B429, 145 (1998).

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

[35] S. Jadach and B. F. L. Ward, Phys. Lett. B274, 470 (1992).

[36] S. Jadach, B. F. L. Ward, and Z. Wąs, Comput. Phys. Commun. 79, 503 (1994).

[37] B. F. L. Ward, Phys. Rev. D36, 939 (1987), ibid. 42 (1990) 3249.

[38] K. Hagiwara and D. Zeppenfeld, Nucl. Phys. B274, 1 (1986).

[39] F. A. Berends et al., Nucl. Phys. B206, 61 (1982).

[40] T. Kinoshita, J. Math. Phys. 3, 650 (1962).

[41] T. D. Lee and M. Nauenberg, Phys. Rev. 133, 1549 (1964).

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

[43] S. Jadach, E. Richter-Was, B. F. L. Ward, and Z. Was, Comput. Phys. Commun. 70, 305 (1992).
[44] S. Jadach, M. Skrzypek, and B. Ward, Phys. Lett. B257, 173 (1991).

[45] S. Jadach et al., Comput. Phys. Commun. 102, 229 (1997).

[46] S. Jadach, W. Płaczek, and B. F. L. Ward, Phys. Lett. B390, 298 (1997).

[47] S. Jadach, B. F. L. Ward, and Z. Wąs, Coherent Exclusive Exponentiation for precision Monte Carlo Calculations, preprint CERN-TH/2000-087; UTHEP-99-0901; hep-ph/0006359; Phys. Rev. D., in press.

[48] S. Jadach, B. F. L. Ward, and Z. Wąs, Comput. Phys. Commun. 130, 260 (2000).

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

[50] S. Jadach, Practical guide to Monte Carlo, 1999, eprint physics/9906056, also available from http://home.cern.ch/~jadach (unpublished).

[51] G. J. van Oldenborgh and J. A. M. Vermaseren, Z. Phys. C46, 425 (1990).

[52] R. Decker, S. Jadach, J. H. Kühn, and Z. Wąs, Comput. Phys. Commun. 76, 361 (1993).

[53] T. Sjöstrand and M. Bengtsson, Comput. Phys. Commun. 43, 367 (1987).

[54] P. Golonka, E. Richter-Wąs, and Z. Wąs, The TAUOLA-PHOTOS-F environment for versioning the TAUOLA and PHOTOS packages, hep-ph/0009302.

[55] T. Pierzchała, E. Richter-Wąs, Z. Wąs, and M. Worek, Spin effects in \( \tau \)-lepton pair production at LHC, preprint TPJU 2/2001, January 2001; hep-ph/0101311.

[56] S. Jadach, The FORTRAN code GlibK, 1995 (unpublished).