KKMCee: Multiphoton MC for lepton and quark pair production at lepton colliders

S.A. Yost

The Citadel

July 19, 2024

with B.F.L. Ward (Baylor University), Z. Wąs (Institute of Nuclear Physics IFJ-PAN) and A. Siodmok (Jagiellonian University)

And acknowledging the essential contributions of S. Jadach, who is responsible for much of this content.

Introduction to KKMCee

- KKMC was developed for precision Z boson phenomenology in e^+e^- collisions, including exponentiated multi-photon effects:
$$e^+e^- \to Z/\gamma^* \to f\bar{f} + n\gamma$$
including exact $O(\alpha), O(\alpha^2 L)$ initial and final state photon emission and initial-final interference, with amplitude-level exponentiation ("CEEX") inspired by YFS soft photon exponentiation.
- $O(\alpha)$EW corrections are included via an independent DIZET module that pretabulates a set of EW parameters and form factors.
- Tau decays are supported via a TAUOLA module.
- Collision energies up to 1TeV are supported, with a LEP2 precision tag of 0.2%.
- This talk will focus on the most recent release, KKMCee 5.00, which is a significant upgrade from previous versions.
- Most significantly, KKMCee is a C++ release, while all previous versions have been FORTRAN releases.

CEEX: Amplitude Level Soft Photon Exponentiation

The physics behind KKMC's CEEX exponentiation, which is a resummation of spin amplitudes, was thoroughly documented in its initial release, S. Jadach, et al, Comput. Phys. Commun. 130 (2000) 260 [arXiv:hep-ph/9912214] and references therein.

Without repeating the formalism, it is worth pointing out some of the reasons CEEX is the ideal basis for state-of-the-art precision photonic physics:

- It naturally facilitates the inclusion of initial-final photon interference effects at all orders.
- Narrow resonance effects important, for example, in initial-final interference near the Z pole can be readily included.
- A complete treatment of spin effects for both the beams and final unstable fermions (such as taus) is possible.

KKMCee 5.00: the New C++ Version

KKMC has split into two branches:

1. KKMCee for lepton colliders is a direct descendent of the original KKMC and is the topic of this talk.
2. KKMChh for hadron colliders applies the same photonic physics to quark collisions. It has been used in LHC studies but is not yet publicly released.

- KKMCee 5.00 is publicly available at GitHub: https://github.com/KrakowHEPSoft/KKMCee
- This version is documented in S. Jadach, et al, Comput. Phys. Commun. 283 (2023) 108556 [arXiv:2204/11949].
- Only external libraries, such as TAUOLA and DIZET 6.45 remain in FORTRAN.
- The upgrade to DIZET 6.45 is documented in A. Arbuzov et al, Comput. Phys. Commun. 260 (2020) 107734 [arXiv:2007.07964].
- DIZET is no longer called directly from KKMCee but is run in advance to create EW tables.
- A Les Houches interface is included to facilitate quark pair hadronization.
- Various models of beam energy spread are included for FCC and ILC/CLIC.
- A new semianalytical tool KKeeFoam is included for cross-checks with analytical resummation.

Improvements in the C++ version 5.00

- The underlying MC algorithm has been replaced by a general-purpose adaptive MC generator, FOAM.

  FOAM generates the energy spread of both beams according to an arbitrary distribution, the total energy of the initial state radiation (ISR), and the type and scattering angle θ_f of the final-state fermion.

- The auxiliary program KKsem for comparing KKMC results with semi-analytical formulas is now replaced with the much more powerful KKfoam tool based on FOAM supporting resummed initial-final interference (IFI) in addition to ISR and final-state radiation (FSR).

- The code is now slimmer, because ROOT provides many services like random numbers, Lorentz kinematics, histogramming, and graphics, formerly handled by the KKMC code.

  A persistency mechanism is now implemented using ROOT, providing more flexibility in the generation and analysis of the MC events. The entire MC generator object may be stored to disk and reread later.

- The handling and database of input parameters is now more transparent and systematic thanks to use of C++ class objects.

- The most advanced and sizable CEEX QED matrix element code is now more compact and transparent thanks to introduction of the auxiliary C++ classes.

- The HEPMC3 event record is instrumental for interfacing to the latest version of PHOTOS and will facilitate the interfacing of KKMCee to modern parton shower MCs and to detector simulation of the collider experiments.

FOAM Event Generation Algorithm

FOAM is a general-purpose adaptive Monte Carlo program able to integrate an arbitrary multi-dimensional distribution provided by the user.

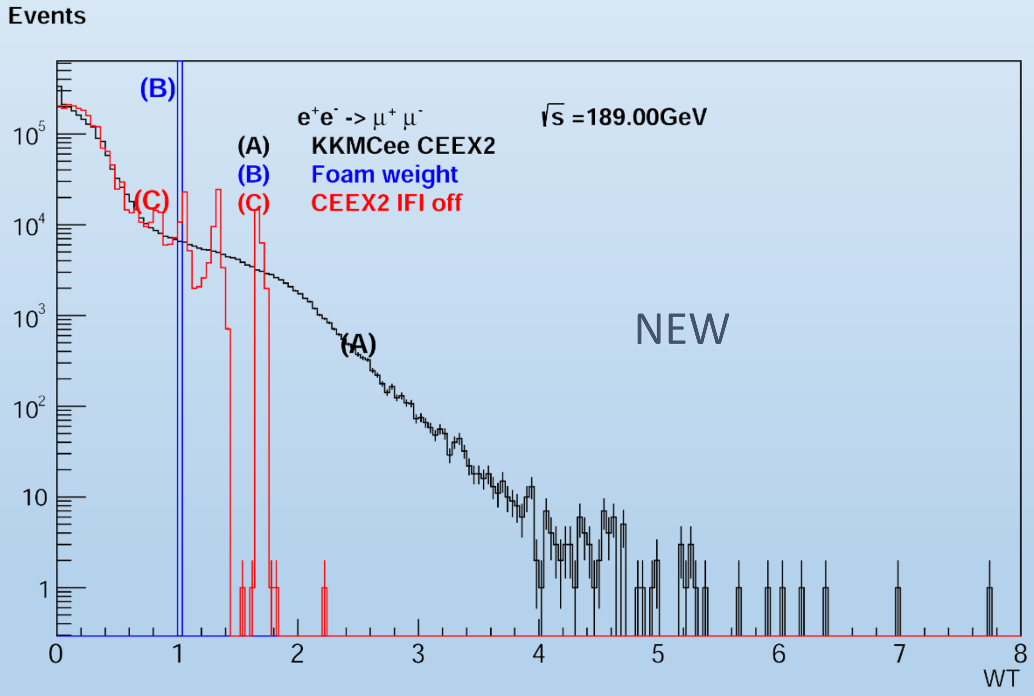
FOAM begins by exploring and memorizing the user distribution in a short exploratory MC run, before its actual use in event generation. During this exploration, it creates a grid of rectangular cells, denser in the regions where the distribution requires it.

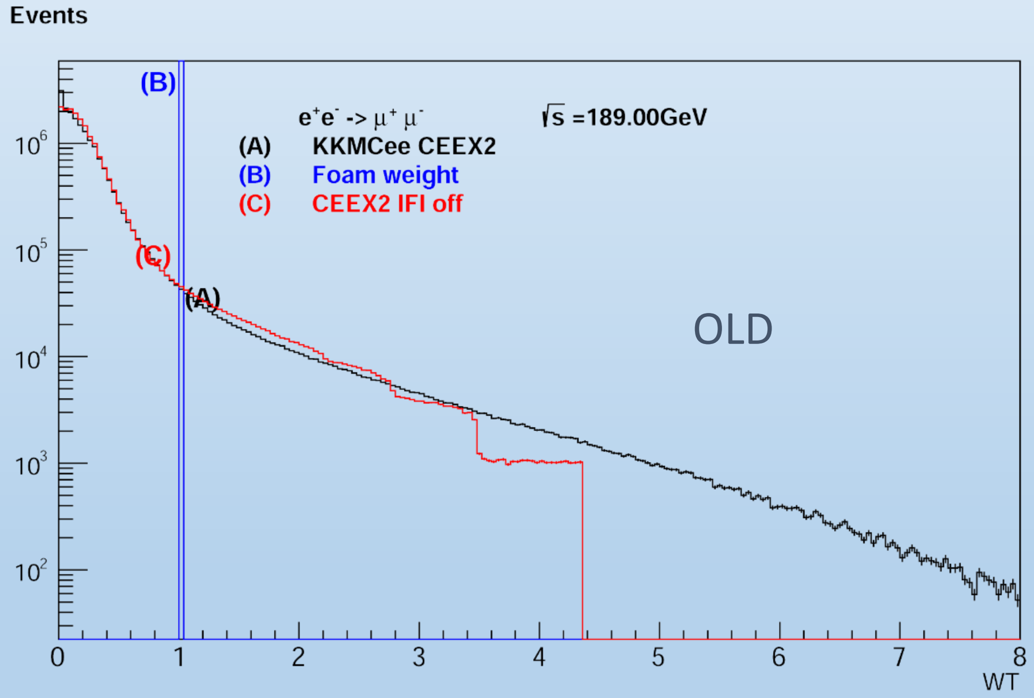
FOAM can efficiently find and account for peaks in the provided distribution automatically. KKMCee uses some sophisticated variable mappings to further improve the efficiency of the MC generation.

FOAM generates the ISR energy fraction v, beamstrahlung parameters z_1 and z_2 ($z = E/\bar{E}$), the type of final fermion $f = \mu, \tau, \nu_e, \nu_\mu, \nu_\tau, u, d, s, c\ b$, and the angle θ_f of the final fermions with respect to the beams.

Improved Weight Distribution via FOAM

Using FOAM to generate θ_f greatly improves the dispersion of MC weights, leading to a more efficient generation of weight 1 events or a smaller variance for weighted events. The graphs below compare weight distributions for 2 million KKMCee 5.00 events on the left to 18 million events from the FORTRAN version on the right for $e^+e^- \to \mu^+\mu^- + n\gamma$ at $\sqrt{s} = 189$ GeV.

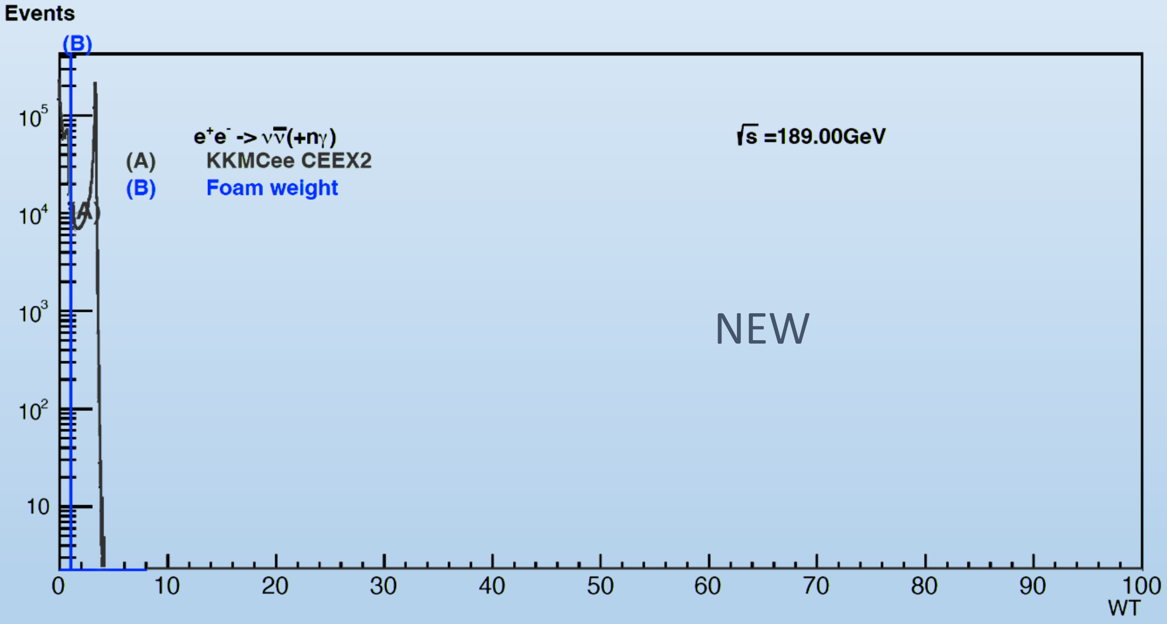


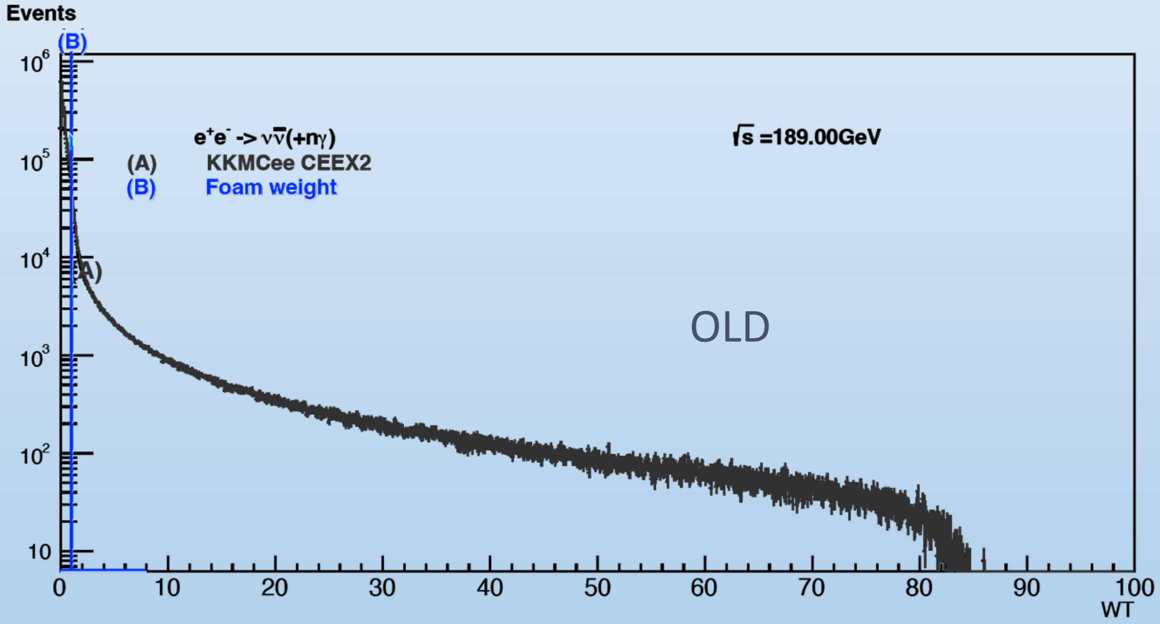


The **black curves (A)** are the complete CEEX2 $O(\alpha^2)$ MC weight, while the **red curves (C)** are the weights with IFI switched off. FOAM's internal weight was set to 1 (**B**).

Improved Weight Distribution for ν_e Events

The effect is most dramatic for ν_e generation above 105 GeV, where a strong peak at $\theta_f = 0$ begins to form due to t-channel W exchange. The graphs below compare the same weight distributions for for $e^+e^- \to \nu_e\bar{\nu}_e + n\gamma$ at $\sqrt{s} = 189$ GeV. There is a factor of 20 improvement in the weight distribution!



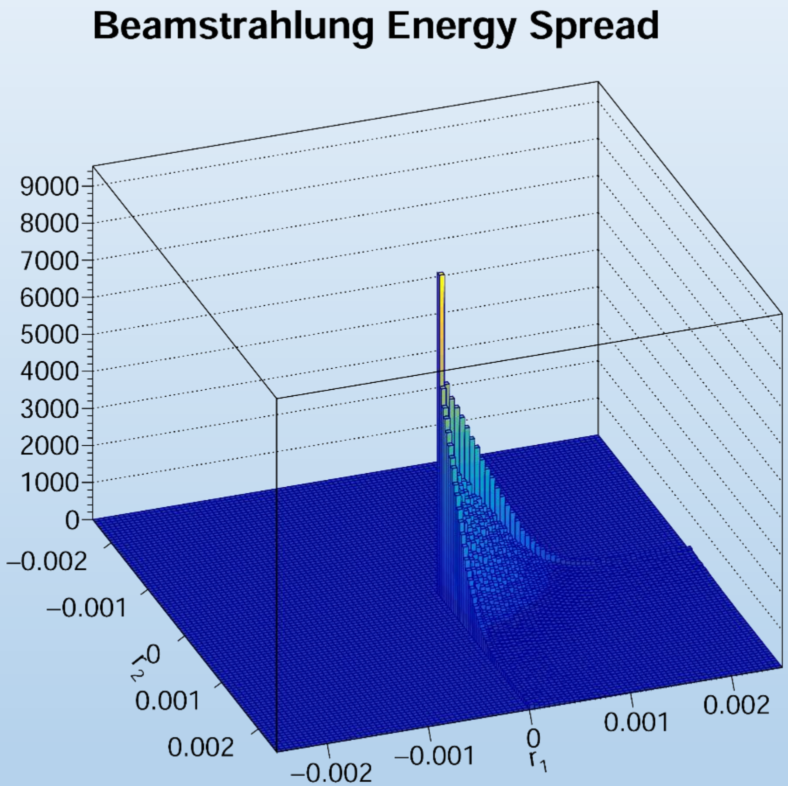


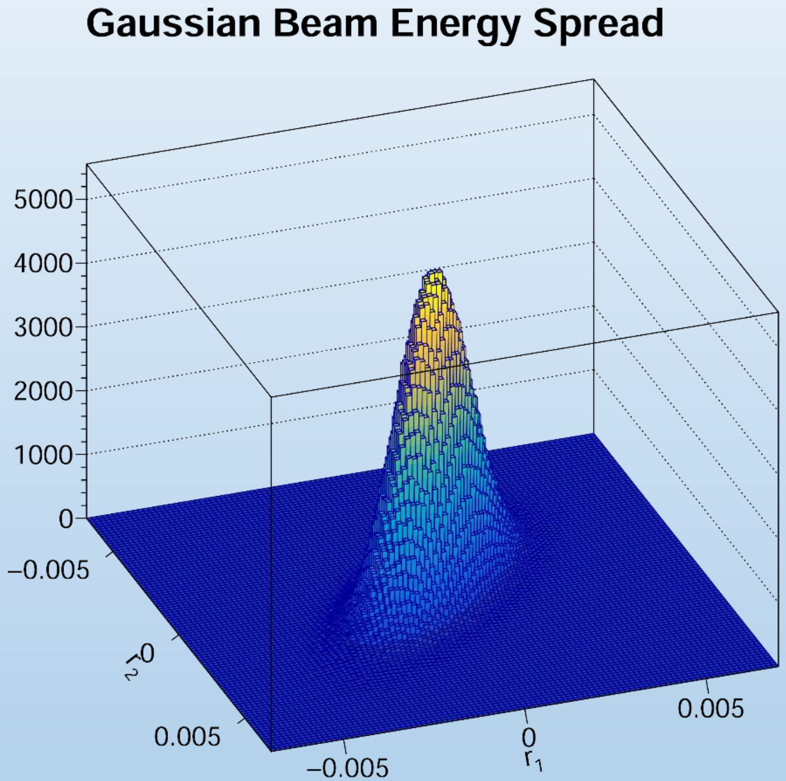
The **black curves (A)** are the CEEX $O(\alpha^2)$ weights. FOAM's internal weight was set to 1 (**B**).

FOAM Beam Energy Spread Distributions

KKMC has long supported beamstrahlung beam spread (BST) via variables $z_i = E_i/\bar{E}_i$ generated using a CIRCE1 parametrization, which has sharp peaks at $z_i = 1$. This is now implemented in FOAM, and an upgrade to CIRCE2 is anticipated.

KKMC now optionally includes Gaussian beam energy spread (BES). The flexibility of FOAM allows the user to provide a combined BES + BST distribution or other custom distribution.





Beam energy spread distributions in $r_i = 1 - z_i$

KKeeFoam Semianalytical Tool

KKeeFoam is now included in the public distribution of KKMCee, replacing the older KKsem.

KKeeFoam is described in detail in S. Jadach and S. Yost, Phys. Rev. D100 (2019) 013002 [arXiv:1801.08611].

KKeeFoam provides the invariant mass $M_{f\bar{f}}$ and angle θ_f of the final fermion pair and internally generates the total longitudinal momentum of the ISR and FSR photons, all integrals over transverse photon degrees of freedom have been done analytically, as has the sum over photons to all orders, in a semi-soft approximation.

KKeeFoam adds IFI, which was not included in KKsem, using two new variables in which the distribution is not positive.

KKeeFOAM also allows separate switching of ISR, FSR and IFI, and IFI may be included even when ISR and FSR are not.

Benchmarks

KKMCee 5.00 reproduces exactly all of previous KKMC benchmarks from the 1999 EW workshop and S. Jadach, Z. Wąs and B. Ward, PRD 63 (2001) 113009 [arXiv:hep-ph/0006359].

The standard muon pair benchmark table at $\sqrt{s} =$ 189 GeV is shown here for 700M unweighted events (top) and compared to the original PRD63 table III (bottom).

Note the 5-digit precision of the new table.

2023

v_{max}	eeFoam $_{IFIoff}$	KKMCee EEX2	CEEX2 IFIoff	CEEX2 IFI on	eeFoam $_{IFIon}$
			$\sigma(v_{max})$ [pb]		
0.02	1.8916 ± 0.0002	1.8921 ± 0.0001	1.8922 ± 0.0001	1.9894 ± 0.0001	1.9907 ± 0.0002
0.10	2.5193 ± 0.0002	2.5200 ± 0.0001	2.5201 ± 0.0001	2.6015 ± 0.0001	2.6029 ± 0.0003
0.30	3.0611 ± 0.0002	3.0615 ± 0.0001	3.0618 ± 0.0001	3.1236 ± 0.0001	3.1224 ± 0.0003
0.50	3.3743 ± 0.0002	3.3737 ± 0.0001	3.3750 ± 0.0001	3.4250 ± 0.0001	3.4194 ± 0.0003
0.70	3.7218 ± 0.0002	3.7185 ± 0.0001	3.7232 ± 0.0001	3.7641 ± 0.0001	3.7500 ± 0.0003
0.90	7.1387 ± 0.0003	7.0998 ± 0.0001	7.1553 ± 0.0001	7.1849 ± 0.0001	7.1495 ± 0.0004
0.99	7.6132 ± 0.0003	7.5604 ± 0.0001	7.6302 ± 0.0001	7.6596 ± 0.0001	7.6233 ± 0.0004
			$A_{FB}(v_{max})$		
0.02	0.5657 ± 0.0001	0.5657 ± 0.0001	0.5657 ± 0.0001	0.6062 ± 0.0001	0.6029 ± 0.0001
0.10	0.5665 ± 0.0001	0.5666 ± 0.0000	0.5666 ± 0.0000	0.5931 ± 0.0001	0.5893 ± 0.0001
0.30	0.5694 ± 0.0001	0.5693 ± 0.0000	0.5692 ± 0.0000	0.5864 ± 0.0000	0.5819 ± 0.0001
0.50	0.5745 ± 0.0001	0.5743 ± 0.0000	0.5742 ± 0.0000	0.5870 ± 0.0000	0.5821 ± 0.0001
0.70	0.5864 ± 0.0001	0.5856 ± 0.0000	0.5857 ± 0.0000	0.5953 ± 0.0000	0.5906 ± 0.0001
0.90	0.3106 ± 0.0000	0.3117 ± 0.0000	0.3097 ± 0.0000	0.3174 ± 0.0000	0.3129 ± 0.0001
0.99	0.2851 ± 0.0000	0.2869 ± 0.0000	0.2845 ± 0.0000	0.2917 ± 0.0000	0.2867 ± 0.0000

2001

v_{max}	$\mathcal{KK}$sem Refer.	$\mathcal{O}(\alpha^3)_{EEX3}$	$\mathcal{O}(\alpha^2)_{CEEX}$ intOFF	$\mathcal{O}(\alpha^2)_{CEEX}$	KORALZ	KORALZ Interf.
			$\sigma(v_{max})$ [pb], $\mathcal{KK}$ M.C. and KORALZ 1-st order			
.01	$1.6712 \pm .0000$	$1.6687 \pm .0020$	$1.6690 \pm .0020$	$1.7679 \pm .0024$	$.9639 \pm .0009$	$.1610 \pm .0009$
.10	$2.5198 \pm .0000$	$2.5164 \pm .0023$	$2.5170 \pm .0023$	$2.5967 \pm .0027$	$2.1919 \pm .0010$	$.0880 \pm .0010$
.30	$3.0616 \pm .0000$	$3.0565 \pm .0024$	$3.0581 \pm .0024$	$3.1190 \pm .0029$	$2.7690 \pm .0010$	$.0545 \pm .0010$
.50	$3.3747 \pm .0000$	$3.3682 \pm .0025$	$3.3713 \pm .0025$	$3.4203 \pm .0029$	$3.0565 \pm .0010$	$.0385 \pm .0010$
.70	$3.7225 \pm .0000$	$3.7131 \pm .0025$	$3.7200 \pm .0025$	$3.7596 \pm .0030$	$3.3649 \pm .0010$	$.0246 \pm .0010$
.90	$7.1434 \pm .0000$	$7.0904 \pm .0024$	$7.1496 \pm .0024$	$7.1789 \pm .0030$	$6.3558 \pm .0010$	$.0210 \pm .0010$
.99	$7.6145 \pm .0000$	$7.5511 \pm .0024$	$7.6254 \pm .0024$	$7.6542 \pm .0029$	$6.7004 \pm .0010$	$.0213 \pm .0010$
			$A_{FB}(v_{max})$, $\mathcal{KK}$ M.C. and KORALZ 1-st order			
.01	$.5654 \pm .0000$	$.5650 \pm .0014$	$.5650 \pm .0014$	$.6111 \pm .0016$	$.5765 \pm .0013$	$.1201 \pm .0013$
.10	$.5664 \pm .0000$	$.5660 \pm .0011$	$.5660 \pm .0011$	$.5922 \pm .0012$	$.5784 \pm .0006$	$.0324 \pm .0006$
.30	$.5692 \pm .0000$	$.5687 \pm .0009$	$.5686 \pm .0009$	$.5856 \pm .0011$	$.5818 \pm .0005$	$.0164 \pm .0005$
.50	$.5744 \pm .0000$	$.5738 \pm .0009$	$.5737 \pm .0009$	$.5863 \pm .0010$	$.5868 \pm .0005$	$.0112 \pm .0005$
.70	$.5864 \pm .0000$	$.5852 \pm .0008$	$.5852 \pm .0008$	$.5947 \pm .0009$	$.5972 \pm .0004$	$.0078 \pm .0004$
.90	$.3105 \pm .0000$	$.3115 \pm .0004$	$.3096 \pm .0004$	$.3170 \pm .0005$	$.3260 \pm .0002$	$.0037 \pm .0002$
.99	$.2851 \pm .0000$	$.2867 \pm .0004$	$.2843 \pm .0004$	$.2912 \pm .0004$	$.3039 \pm .0002$	$.0024 \pm .0002$

Future Developments

The C++ version is intended to provide a flexible basis for future development of KKMCee.

Examples of anticipated upgrades are…

- Adding CEEX O($\alpha^3 L^3$) corrections, at least, which are already present in EEX. Possibly higher-order leading logs may be added as well.
- Forcing a visible photon at the generator level in the neutrino pair channels.
- Automatic construction of the CEEX matrix element (~ 4000 lines of code) for porting to other processes such as HZ production and decay.
- A new $O(\alpha^2)$ EW library. But note that 1-loop EW corrections for $e^+e^- \to f\bar{f}\gamma$ are still unavailable for fully differential distributions.
- A more efficient algorithm for some corners of phase space, such as 2 very hard photons.
- Integrating the Bhabha process into KKMCee?