Prof. Dr. Stefan Kehrein
Prof. Dr. Marcus Müller
Prof. Dr. Steffen Schumann

Summer term 2024

Advanced Computational Physics Lab

Programming project 3: Monte Carlo Simulations for Particle Physics

Tutor:
Enrico Bothmann, enrico.bothmann@uni-goettingen.de

Q & A sessions:
Wed 2:15pm–5:45pm & Fri 10:15am–11:45am

At the end of this project, you will have built a Monte Carlo event generator for the simulation of electron-positron annihilation and the subsequent production of a quark-antiquark pair, see Fig. 1. We will supplement the fixed-order description of the $2 \to 2$ scattering process with an all-order simulation of the cascade of bremsstrahlung emitted by the quark-antiquark pair. The first part of the project, Exercise 1.1, will deal with the $2 \to 2$ process, whereas the second part, Exercise 1.2, is about the bremsstrahlung cascade. The combination of both will give us a fairly realistic simulation that can be used to predict a number of physics observables which can be measured at a electron–positron collider such as the Large Electron–Positron Collider (LEP), which has been in operation at CERN from 1989 until 2000 and still is the most powerful human-made lepton accelerator in history.

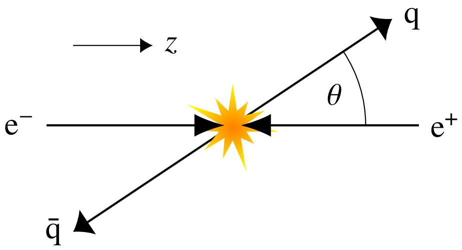


Figure 1: Illustration of the $\mathrm{e}^+\mathrm{e}^- \to \mathrm{q}\bar{\mathrm{q}}$ scattering process. The z-axis is oriented along the e^- beam direction. The scattering angle θ spans between the z axis and the outgoing quark line.

Exercise 1.1: Simulation of the $\mathrm{e}^+\mathrm{e}^- \to \mathrm{q}\bar{\mathrm{q}}$ scattering process at fixed-order QED

For the first bit, the electron-positron annihilation and subsequent production of a quark-antiquark pair, we will use the squared leading-order matrix element $|\mathcal{M}_{\mathrm{q}\bar{\mathrm{q}}}|^2$ for the production of a given quark flavour q, which can be calculated in perturbative quantum field theory. At leading order, the scattering is mediated by virtual photons and Z bosons, cf. App. A, where the matrix element is given, along with its representation in terms of Feynman diagrams. The probability distribution for the process to occur, allowing for any light quark flavour q to be produced, is proportional to the differential cross section, which is given by

$$\frac{\mathrm{d}\sigma}{\mathrm{d}s\mathrm{d}(\cos\theta)\mathrm{d}\phi} = f_{\mathrm{conv}} f(s) \frac{1}{8\pi}\frac{1}{4\pi}\frac{1}{2s}\sum_{\mathrm{q}=1}^{N_{\mathrm{q}}} |\mathcal{M}_{q\bar{q}}(s, \cos\theta, \phi)|^2,$$

where $f(s)$ is a normalised distribution that defines the beam spectrum, and N_{q} is the number of light quark flavours that can be produced, which is listed among the other input values of the calculation in App. B. There, you will also find the value for the conversion factor f_{conv}, which must be used to give cross sections in physical units of picobarns (pb). To calculate the total cross section σ, one needs to integrate over the kinematic phase-space quantities s, $\cos\theta$ and ϕ, which are defined in App. A.

a) (**Code: 7 points**) Write a program to integrate the cross section numerically using Monte Carlo event generation, for the case that the beam energies are fixed such that $s = M_{\mathrm{Z}}^2$, i.e. $f(s) = \delta(s - M_{\mathrm{Z}}^2)$.

   Note that you can either evaluate the sum over outgoing quark flavours q explicitly, or you can use $\sum_{\mathrm{q}} |\mathcal{M}|^2 = N_f \langle|\mathcal{M}|^2\rangle_{\mathrm{q}}$ and evaluate the average $\langle\ldots\rangle_{\mathrm{q}}$ by picking a random flavour q for each Monte Carlo point. Implement both of these options.

   In this exercise and the remaining ones, make sure that your code always produces the same results by explicitly setting a fixed seed for your random number generator. This makes debugging easier and ensures that the tutor will be able to reproduce your results exactly.

   Hint: You do not need to implement a Hit-or-Miss Monte Carlo, neither in this exercise nor the remaining ones. Simply use the basic procedure for a "spatial" problem given on slide 10 of the first lecture. The given formulae for the estimates of the mean and the error trivially generalise to the multi-dimensional case. Also the technique of importance sampling is not required for now, we will ask for it explicitly below, in part f).

b) (**6 points**) First, confirm that the Monte Carlo estimates for σ are statistically equivalent for both flavour summing options. In the following, only use the random-flavour option.

   Check that your cross section agrees with 42 250(50) pb within the given statistical accuracy.[1] Plot the Monte Carlo error estimate for an increasing number of sample sizes N, e.g. $N = 10^1 \ldots 10^4$. What shape do you find in a log-log plot? Does that agree with the expectation of the Monte Carlo error dependence on N?

c) (**Code: 7 points**) At realistic collider setups, the beam spectrum is not entirely sharp, because the incoming leptons tend to emit photons before colliding. This is called beamstrahlung. To keep it simple, let us make the (unphysical) assumption that s is distributed uniformly between $(M_{\mathrm{Z}} - 3\Gamma_{\mathrm{Z}})^2$ and $(M_{\mathrm{Z}} + 3\Gamma_{\mathrm{Z}})^2$. To take this into account, extend your Monte Carlo integrator to also sample s and calculate the total cross section for the flat beam spectrum. Bin the weight of each Monte Carlo point in an s-histogram. For the histogram, you can use the provided `Histo1D` python class, which implements a simple histogram that can be filled with weighted Monte Carlo events.

[1] Note that all of our reference results have also been generated by Monte Carlo integration. Therefore, they come with an uncertainty given by the estimate of the standard deviation.

d) (**9 points**) Confirm that your Monte Carlo estimate for σ with the flat beam spectrum is compatible with 9880(50) pb. Why is it smaller than the result in a)/b)? Plot the Monte Carlo error against N in a log-log plot as in b). Plot the s histogram. Check that your result is consistent with your fixed-s Monte Carlo implementation by re-doing the integration of a) for different s_0 in $f(s) = \delta(s - s_0)$, thus "scanning" over the range of s used in c). Plot the resulting line on top of the histogram plot. This histogram and the line should be compatible with each other after dividing the histogram by the flat $f(s)$ you used in c). Confirm that this is the case and explain why the histogram must be divided by $f(s)$ to find agreement.

e) (**6 points**) Load the `vegas` package for python which implements a Monte Carlo integrator that automatically adapts to the integrand, and apply it to the problem stated in c). Use 10 iterations with 1000 Monte Carlo points each and print a summary of the VEGAS integration by using the `summary()` method of the integration's return value. Confirm that the Monte Carlo error estimate is reduced after each iteration, as an effect of the VEGAS optimisation. Use the `map.show_grid()` method to show the adapted integration grids, in particular the one that corresponds to the $(\cos\theta)$–s plane. Comment on the behaviour of the grids; for which dimensions is a non-trivial binning visible?

f) (**Code: 10 points**) The shape of the integrand for $s \approx M_Z^2$ is mostly determined by the Breit-Wigner function $g(s) = ((s - M_Z^2)^2 + M_Z^2\Gamma_Z^2)^{-1}$, i.e. the denominator of $\chi_2(s)$ (this is because, relatively speaking, the numerator is changing only mildly around M_Z^2, and other contributions are much smaller in this region compared to the χ_2 one). Extend your integrator for the flat beam spectrum to use importance sampling with the Breit-Wigner mapping as discussed in the lecture to improve your sampling efficiency.

g) (**6 points**) Confirm that your integration result agrees with your result from c)/d) and compare the Monte Carlo error estimate with and without importance sampling after the same number of points N.

Note that in real high energy physics applications we would usually have multiple peak structures. Therefore, a sum of importance sampling mappings is used, which is automatically determined from the structure of the process. To draw a random point, one of those mappings—or channels—is randomly selected. This is called a multi-channel importance sampling. In addition, each channel is usually composed with a VEGAS optimisation, in order to adapt to the less singular behaviour of the integrand not captured by the channel mapping (e.g. the $\cos\theta$ dependence, for which you can see a slight adaptation in your grids in e)).

Exercise 1.2: Simulation of the bremsstrahlung cascade at all-order QCD

With the 2 → 2 scattering set up, we now turn to the simulation of additional QCD bremsstrahlung produced by the final-state quark–antiquark pair. This stage can also be simulated using perturbative methods, although in a special kinematic limit where such emissions are strongly enhanced. This restriction allows us to include terms from all orders in perturbation theory, which yields a good approximation for the bulk of the radiation and hence an understanding of how the quarks evolve, giving rise to cone-shaped cascades of quarks and gluons, cf. Fig. 2.

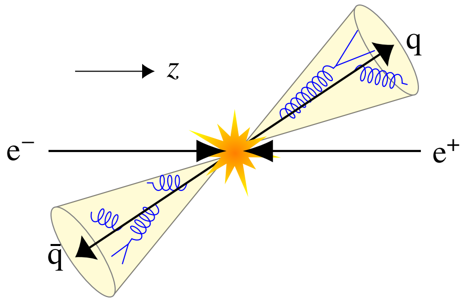


Figure 2: Illustration of the $e^+e^- \to q\bar{q}$ scattering process as in Fig. 1, but this time additional bremsstrahlung emissions off the quark lines are depicted (blue), leading to cone-shaped jets of particles (light yellow).

Quarks and gluons are collectively referred to as partons, and the cone-shaped cascades are also called (partonic) jets. The kinematic limit mentioned above corresponds to the emitted parton being collinear with its mother parton (i.e. the emission angle θ is small) and/or soft (i.e. the daughter's energy E is small). In this case, each emission can be treated independently from all previous emissions, since the cross section completely factorises into a part without the emission, and a universal emission factor, such that an iterative algorithm can be applied, treating the subsequent emissions as a Markov Chain process. The numerical simulation of this procedure is called a "parton shower". In the following exercises, you can again assume $f(s) = \delta(s - m_Z^2)$, i.e. we generate events at the Z mass peak.

a) (**8 points**) Whereas the lowest-order matrix element depends on the QED coupling α only, the interaction between quarks and gluons is determined by the QCD coupling (or strong coupling) α_s. This coupling depends significantly on the energy scale of the interaction. Instantiate an object of the provided `AlphaS` class. As an input, use M_Z and $\alpha_s(M_Z)$.[2] As usual, numerical values can be found in App. B. Use the `AlphaS` object to plot the coupling for $t = 1\,\text{GeV}^2 \ldots 10\,000\,\text{GeV}^2$, using a logarithmic scale for the x axis. How does the coupling behave for small scales, and what does this mean for a perturbative expansion around $\alpha_s \ll 1$?

b) (**Code: 7 points**) There is no time to code the shower algorithm as presented in the lecture from scratch, we therefore provide a `Shower` python class. Have a look inside to get a rough overview of its parts. The remaining task now is to hook up the shower as an afterburner to our 2 → 2 Monte Carlo simulation. Start by using the provided four-vector class `Vec4` and set up the four-momenta for the incoming and outgoing particles for each evaluation of your cross section calculator, using s, $\cos\theta$ and ϕ. These momenta and the flavour q of the outgoing quarks[3] will be needed as an input to the parton shower algorithm. The other input needed is the colour charge of the outgoing quarks. Due to the conservation of the charge, the quark will carry a colour, whereas the anti-quark carries the corresponding anti-colour. Use the `Particle` class and generate a list of the four particles. Note that `Particle` uses a pair of integers to store colour information. The first entry stores the colour, and the second the anti-colour of the particle. This is because gluons carry both colour and anti-colour at the same time. The integer `0` is used to indicate the absence of colour or anti-colour. Hence, colourless particles use `[0, 0]`, quarks `[n, 0]`, anti-quarks `[0, n]` and gluons `[n, m]`, where $n, m > 0$ are integers. Use $n = 1$ for the initial quark-antiquark pair. Besides the colour information, the `Particle`

[2] Only the evolution of the coupling can be calculated perturbatively, which means that a measurement of the coupling at some energy scale, such as M_Z, must be provided as a starting point for the evolution.

[3] Therefore, you need to use flavour sampling instead of explicit summing, such that you have a definite flavour state for each Monte Carlo scattering event.

constructor also requires information about the type of particle it represents. This type is encoded by a number, see App. C. Having created the list of particles, you can fill the first argument of the `run` function of your `Shower` object. The second, `t`, denotes the starting scale of your shower. This is given by the energy scale of the $2 \to 2$ scattering, i.e. in this case $\mathtt{t} = s = M_Z^2$.

c) (**6 points**) Run 1000 events and report the average number of final-state particles after showering, taking into account the Monte Carlo weight of each event. Confirm that your result is compatible with 5.2(2).

d) (**Code: 9 points**) Because the probability to emit an additional quark or gluon for small angles or small energies diverges, quantities like the number of particles after showering are ill-defined and depend on unphysical parameters like the shower cut-off parameter t_0. Also, experiments have a limited resolution, such that the emission of collinear or soft partons will not change the experimental signature. Jet algorithms cluster particles into jets that are well-separated to address both issues and allow for well-defined comparisons of experimental data and theory calculations. Resolve the `TODO` in the `Analysis` python class by including the Durham jet algorithm discussed in the lecture.

e) (**9 points**) Instantiate an `Analysis` object and feed it with events from your event loop, by calling `analyze(event, weight)`, where `event` is the list of particles after showering, and `weight` is the event weight. After generating all events, you must in addition call `finalize` to wrap up the analysis, and to write the produced histograms into a file. The analysis histograms differential jet rates, i.e. the distance scale at which a $(n+1)$-jet event starts to look like an n-jet event. These rates are sensitive to the emission structure of QCD and are often used to check how well QCD predictions describe collider data. Finally, you can use the `plot_jet_histograms` function to compare your results with the ones provided in the file `sherpa.yoda`. This file has been generated by SHERPA, a general-purpose Monte Carlo event generator that is developed by our Göttingen group in collaboration with other groups in Europe and the US. Confirm that the two predictions are compatible.

As you can see, it is not too hard to reproduce results of a tool widely used for predictions compared to measurements at the LHC at CERN! Admittedly, things get quite a bit more complicated when one increases the precision of the matrix element and the parton shower calculation, includes the non-perturbative clustering of the partons into hadrons, or generalises the code for all possible Standard Model (and even New Physics) interactions. However, the basic concepts of the Monte Carlo algorithms and the event generation "pipeline" we have covered throughout this project remain largely the same.

In order to facilitate the supervision, please set up a git repository at gitlab.gwdg.de and give reading rights to the tutor.

Criteria for grading:

- (**40 points**) Upload the codes for Exercises 1.1a), 1.1c), 1.1f), 1.2b) and 1.2d) to the git-repo with instructions on how to run them. The codes need to be executable and the tutor needs to be able to verify that the codes produce the data. Include the data shown in the figures as ASCII files and store those in the repository as well.[4] Make

[4] Generally, it is good practice in software engineering **not** to include data in a repository, because its size will rapidly grow—make an exception here.

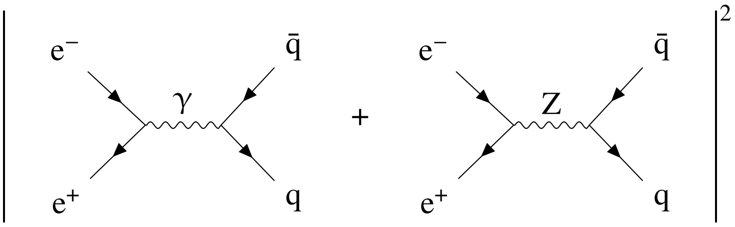
sure that the code is readable and/or well-documented (max. 5 points out of 40 for the code). Also, upload all additional routines used to process data.

- Submit a written report (e.g., using latex, at maximum 10 pages) that contains the figures and answers to the above questions to the tutor via e-mail. This is freeze time: The tutor will pull the code from the repo and no further changes will be accepted.
- (**10 points**) Formal aspects: Proper use of units, preparation of figures and figure captions.
- (**50 points**) Correctness and quality of the answers to the topic specific questions (see distribution in the exercises).

Appendix A: the $e^+e^- \to q\bar{q}$ matrix element

The leading-order calculation for the $e^+e^- \to q\bar{q}$ matrix element yields the following result:

$$|\mathcal{M}_{q\bar{q}}(s, \cos\theta, \phi)|^2 =$$



$$= (4\pi\alpha)^2 N_C \left[\left(1 + \cos^2\theta\right)\left\{Q_e^2 Q_q^2 + 2Q_e Q_q V_e V_\mathrm{q} \chi_1(s) + \left(A_e^2 + V_e^2\right)\left(A_\mathrm{q}^2 + V_\mathrm{q}^2\right)\chi_2(s)\right\} + \cos\theta\left\{4Q_e Q_\mathrm{q} A_e A_\mathrm{q} \chi_1(s) + 8A_e V_e A_\mathrm{q} V_\mathrm{q} \chi_2(s)\right\}\right], \qquad (1)$$

where the first line shows the representation in terms of Feynman diagrams. The kinematics of the process are given by the square of the total energy in the centre-of-mass system, $s = E^2_\mathrm{CMS}$, the cosine of the scattering angle θ between the incoming positron and the outgoing quark, cf. Fig. 1, and the azimuthal angle of the outgoing quark in the plane perpendicular to the beam. The coupling strength of QED is α and N_C gives the number of colour charges the quarks can carry. Q_f is the charge of the particle f. Note that there are five light quark flavours, which are treated massless here, just like the electron. These flavours are the u, d, c, s and b quarks. We also have vector and axial couplings of the fermions to the Z in eq. (1), given by $V_\mathrm{f} = T^3_\mathrm{f} - 2Q_\mathrm{f}\sin^2\theta_W$ and $A_\mathrm{f} = T^3_\mathrm{f}$, respectively, where T^3_f is either $+1/2$ or $-1/2$, see App. B, and θ_W is the Weinberg angle that determines the mixing of photons and Z bosons in the unified electroweak theory. The $Q^2_\mathrm{e}Q^2_\mathrm{q}$ term stems from the square of the γ amplitude, the χ_2 one from the square of the Z amplitude and the χ_1 from the photon-Z interference. They are defined as follows:

$$\chi_1(s) = \kappa \frac{s\left(s - M_\mathrm{Z}^2\right)}{\left(s - M_\mathrm{Z}^2\right)^2 + \Gamma_\mathrm{Z}^2 M_\mathrm{Z}^2},$$

$$\chi_2(s) = \kappa^2 \frac{s^2}{\left(s - M_\mathrm{Z}^2\right)^2 + \Gamma_\mathrm{Z}^2 M_\mathrm{Z}^2},$$

where the mass and the width of the Z boson are given by M_Z and Γ_Z, respectively. The prefactor is given by

$$\kappa = \frac{1}{4\sin^2\theta_W(1 - \sin^2\theta_W)}.$$

Appendix B: Numerical values

For your implementation, please use the numerical values listed in Tab. 1 for the quantities defined above. Note that masses and decay widths are given in GeV. However, for the purpose of the implementation, all quantities are considered to be dimensionless, and the conversion factor f_{conv} can be used to cast results for cross sections in the conventional unit of picobarns (pb).

Table 1: Table of quantities needed in the implementation.

Symbol	Value	Description
α	$1/129$	QED coupling
$\alpha_s(M_Z)$	0.118	QCD coupling at the Z mass scale
M_Z	91.2	Z boson mass
Γ_Z	2.5	Z boson decay width
$\sin^2\theta_W$	0.223	squared sine of the Weinberg angle
Q_e	-1	electric charge of the electron
$Q_{u,c}$	$2/3$	electric charge of (light) up-type quarks
$Q_{d,s,b}$	$-1/3$	electric charge of down-type quarks
$T^3_{u,c}$	$1/2$	weak isospin of (light) up-type quarks
$T^3_{e,d,s,b}$	$-1/2$	weak isospin of down-type quarks and the electron
N_q	5	number of light quark flavours (i.e. u, d, s, c and b)
N_C	3	number of QCD colours
f_{conv}	$3.89379656 \cdot 10^8$	physical units conversion factor

Appendix C: Particle numbering convention

In the provided library code, you will find that particle species are referred to by specific numbers. These numbers are defined in a convention called the “Monte Carlo Particle Numbering Scheme”, which can be found in the Review of Particle Physics. In Tab. 2, we reproduce the numbers relevant for the project as a reference. If a particle has a distinct antiparticle, that antiparticle is referred to by the negative number, e.g. -11 stands for a positron, e^+.

Table 2: Particle numbering scheme.

Name	Symbol	Number
down quark	d	1
up quark	u	2
strange quark	s	3
charm quark	c	4
bottom quark	b	5
electron	e^-	11
gluon	g	21

Literature:

- S. Weinzierl, “Introduction to Monte Carlo methods”, NIKHEF-00-012, arXiv:hep-ph/0006269 [hep-ph] (`https://inspirehep.net/literature/529256`)
- A. Buckley et al., “General-purpose event generators for LHC physics”, Phys. Rept. **504** (2011), 145-233 (`https://inspirehep.net/literature/884202`)
- M. Tanabashi et al. [Particle Data Group], “Review of Particle Physics,” Phys. Rev. D **98** (2018) no.3, 030001 (`https://hal.archives-ouvertes.fr/hal-01867148`)