1 Some crosssections

The part that translates the variables that vegas uses to integrate our functions into variables that are needed in the matrix element is called the kinematics part. We have already seen that we can translate the integration over the final states into something that makes some sense (i.e. variables for which we can determine the range).

Now we have to actually do this. The first program would be a two body decay. We have a two particle phase space and we have seen that we have to integrate over the angles in the CM-frame. If the system is axially symmetric we only have to integrate over the azimuthal angle and we can take

$$\cos\theta^{CM} \quad = \quad 1 - 2x_0 \tag{1}$$

in which x_0 is one of the variables that come from vegas. Those variables have values between 0 and 1. Note that this transformation has a Jacobian:

$$\frac{d\cos\theta^{CM}}{dx_0} \quad = \quad -2 \tag{2}$$

Of course the minus sign disappears when we flip the order of the integration boundaries.

If we want to emulate an experiment it is often a good idea to also choose a value for ϕ^{CM}. There are two good reasons for this:

- We can check that there is no dependence on ϕ^{CM}.

- The detector may not be axially symmetric.

Of course, if we are very sure that everything is axially symmetric and no checks are needed we can take a fixed value for ϕ^{CM}. This is usually taken to be zero, which means that the events take place in the x-z plane. When we do the ϕ integral we would have

$$\phi^{CM} \quad = \quad 2\pi x_1 \tag{3}$$

with a Jacobian of 2π.

In the library we have a routine for a generic two body phase space. This routine has the name mgoto2. It assumes that the initial state can be moving with any momentum. This state is represented by a 4-vector. The routine determines the Lorenz transformation from the CM-frame to this moving frame. Then it uses the two input values between 0 and 1 to select a $\cos\theta^{CM}$ and ϕ^{CM}. From this it can determine the outgoing momenta in the CM-frame, after which these momenta are boosted to the frame of the initial state. The return value of the routine is the Jacobian of the transformation.

Routines like mgoto2 can make life very easy. We will see a few more later. The whole library that contains these routines, vegas, the histogramming routines and a few more things is called axolib. The first version of this library dates from 1977 (at that time in Fortran). In the years after it has been extended

a bit. Most (semi)automatic systems that exist nowadays have started out using similar routines and libraries. The current version of axolib is in C.

To use these routines efficiently in a language like C, we need to determine a number of data types. This is done in the file axolib.h. The most important for now is

```
typedef struct {
    double e;
    double px,py,pz;
    double m;
} PVEC;
```

This data type defines a 4-vector in momentum space. It contains however also the mass of the particle. One may argue that this isn't necessary, but try to imagine what happens when we have an electron which moves along the z-direction and has an energy of 100 GeV. If we know the energy and the three momentum we can compute the mass:

$$
\begin{aligned}
m &= \sqrt{e^2 - p^2} \\
  &= \sqrt{(e-p)(e+p)}
\end{aligned} \tag{4}
$$

Now how big is $e - p$? Imagine that we have to compute this from those two numbers. We get

$$
\begin{aligned}
e - p &= \frac{(e-p)(e+p)}{e+p} \\
      &= \frac{m^2}{e+p} \\
      &= \mathcal{O}(10^{-9})
\end{aligned} \tag{5}
$$

which means that we loose about 11 digits, just for something simple. This is why we carry seemingly superfluous information around.

The above cancellation is due to the Lorenz invariance of space-time. Many kinematics routines have to be protected against numerical cancellations and it is the major part of the job in their construction. Also the matrix elements may contain very bad numerical cancellations between the terms. These are called gauge cancellations. Sometimes special variables need to be used to transfer information from the kinematics to the matrix element in a numerically stable way. Another solution to this problem can be to use very high precision in the calculations. This can go at the cost of very much computer time. We may get to see some of these cancellations at a later stage.

Back to our program for the reaction $e^- e^+ \to \mu^- \mu^+$. The structure of such programs is that we have:

- A main routine that contains the definition of some constants, the call to some initialization routines and finally the call to vegas.

- A routine that contains the kinematics. Here we will give that routine a name that starts with the letter f as in feemm. This is the routine that gets called by vegas and its argument is an array of numbers between 0 and 1. The return value is the value of the complete integrant including overal factors.

- A routine that contains the matrix element. This routine is called by the kinematics routine.

- Various library routines like vegas, histogram routines, mgoto2, etc.

And for each project we have a make file. Because this reaction corresponds to the third program for which we use a make file we have named the make file make3 and the file of the main program main3.c.

For the moment the important routines are the first two. Hence you should have a good look at them and try to understand everything that is done there.

Homework: Determine the crosssection of this reaction inside the experimental acceptance of $|\cos\theta^{CM}| < 0.8$ and assuming there is a dead zone in the detector for 10 degrees $< \theta^{CM} < 25$ degrees and ϕ between 17 and 39 degrees. Take the energy to be 20 GeV.

The next reaction we study is the reaction $e^-e^+ \to e^-e^+$. This reaction has two diagrams:
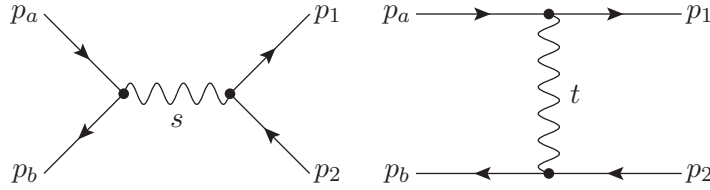
The first diagram has a propagator with the invariant s and hence is called an s-channel graph. The second diagram has a propagator that corresponds to the invariant called t and hence is called a t-channel graph. The complete matrix element is given in the file meeee.c.

In principle the kinematics is the same as in the muon case. There is however one complication. If we compute what the quantity t is in terms of the θ angle in the CM-frame we have:

$$
\begin{align}
t &= (p_a - p_1)^2 \tag{6}\\
&= 2m_e^2 - 2E_aE_1 + 2p_ap_1\cos\theta^{CM} \tag{7}\\
&= 2m_e^2 - 2(s/4) + 2(s - 4m_e^2)/4\cos\theta^{CM} \tag{8}\\
&= -\frac{s - 4m_e^2}{2}(1 - \cos\theta^{CM}) \tag{9}
\end{align}
$$

and we see that at the edge of phase space t can have the value zero. Because in the matrixelement the t-channel graphs introduce terms with t in the denominator, this means that the matrix element goes to infinity when $\cos\theta^{CM}$ goes to one and after integration it is still infinite! This is a well known effect: elastic

electromanetic scattering has an infinite crosssection. The infinite part is however in a region where it is impossible to see the difference between scattering and no scattering. This is the region in the beampipe. There is no detector in that region. Hence we won't observe this. If we only integrate over the region that we do observe the crosssection is finite. This means that experimental cuts are essential here. These are put in the function feeee.c.

If we use the same kinematics as we use for the muons we see already that the crosssection is much bigger. This is also due to the t-channel graph. In general t-channel graphs give much bigger contributions than s-channel graphs. The s-channel graphs give usually crosssections that go asymptotically like $\mathcal{O}(1/s)$ while the t-channel graphs give crossections that go like $\mathcal{O}(\ln s)$ or even some power of the logarithm.

Our program will produce also some distributions and when we look at them we see that almost all the crosssection is in one bin. It takes vegas a bit of work to zoom in on this peak. This is why we have a second routine in which we use a mapping to take care of the $1/t^2$ behaviour. First we notice that if we have a cut c on the value of $\sin\theta^{CM}$ this translates into the value a for the variable $y_0 = (1 - \cos\theta^{CM})/2$ which we feed to the routine mgoto2. The integral we have to take care of is now

$$\int_a^{1-a} dy_0 \; 1/y_0^2 \tag{10}$$

with

$$\begin{aligned} a &= \frac{1 - \sqrt{1 - c^2}}{2} \\ &= \frac{c^2}{2(1 + \sqrt{1 - c^2})} \end{aligned} \tag{11}$$

Now we can proceed with the mapping as explained in the chapter about Monte Carlo integration with

$$\begin{aligned} x_0 &= \frac{F(y_0) - F(y_-)}{F(y_+) - F(y_-)} \\ &= \frac{-1/y_0 + 1/a}{-1/(1 - a) + 1/a} \end{aligned} \tag{12}$$

and hence

$$\begin{aligned} y_0 &= \frac{1}{1/a - x_0 b} \\ b &= 1/a - 1/(1 - a) \\ \frac{dy}{dx} &= b \, y_0^2 \end{aligned} \tag{13}$$

We put this in the routine feeeea and can now run the program prog2a. It becomes very clear that this program converges much faster and in the differential distributions we can see why.

Remark: often there may be terms of the type $1/t^2$ but they are not the dominant terms. The really important terms are the terms in $1/t$ because they have a much bigger coefficient/factor. The $1/t^2$ mapping occurs more in special cases or in integrals over invariant masses (the t-integral is equivalent to an angle integral).

Homework: The third reaction concerns the reaction $e^- e^+ \to \gamma\gamma$. This reaction has two Feynman diagrams and its matrix element is given in the file meegg.c. Using the examples of main3.c, main4.c, main4a.c, feemm.c feeee.c and feeeea.c you are to construct the files main5.c and feegg.c and obtain the crosssections for this process. You should again use an anglecut variable with the value 0.05 and a beam energy of 15 GeV. If you want to create histograms, use the value 5 in the file boundaries.c.

Reaction number 4 is more complicated. We will look at the reaction $e^- e^+ \to \tau^- \tau^+ \to e^- \overline{\nu}_e \nu_\tau \mu^+ \nu_\mu \overline{\nu}_\tau$. This has six particles in the final state!
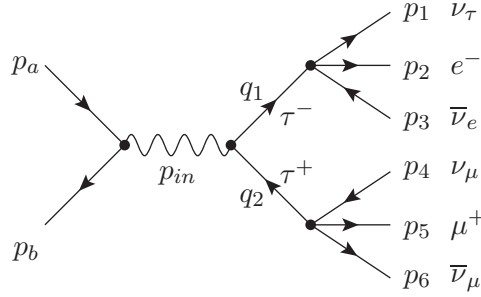
There is a generic way by which any number of particles in the final state can be obtained. Given the phase space integral

$$
\begin{aligned}
P.S. \quad = \quad & \int \frac{d^4 p_1}{(2\pi)^4} (2\pi)\delta(p_1^2 - m_1^2)\theta(E_1) \\
& \int \frac{d^4 p_2}{(2\pi)^4} (2\pi)\delta(p_2^2 - m_2^2)\theta(E_2) \\
& \int \frac{d^4 p_3}{(2\pi)^4} (2\pi)\delta(p_3^2 - m_3^2)\theta(E_3) \\
& \int \frac{d^4 p_4}{(2\pi)^4} (2\pi)\delta(p_4^2 - m_4^2)\theta(E_4) \\
& \int \frac{d^4 p_5}{(2\pi)^4} (2\pi)\delta(p_5^2 - m_5^2)\theta(E_5) \\
& \int \frac{d^4 p_6}{(2\pi)^4} (2\pi)\delta(p_6^2 - m_6^2)\theta(E_6) \\
& \delta^{(4)}(p_a + p_b - p_1 - p_2 - p_3 - p_4 - p_5 - p_6)
\end{aligned}
\tag{14}
$$

one can insert the integrals

$$
\begin{aligned}
1 \quad = \quad & \int \frac{d^4 q_1}{(2\pi)^4} (2\pi)^4 \delta^{(4)}(q_1 - p_1 - p_2 - p_3) \\
& \int \frac{d^4 q_2}{(2\pi)^4} (2\pi)^4 \delta^{(4)}(q_2 - p_4 - p_5 - p_6)
\end{aligned}
\tag{15}
$$

after which the complete integral becomes

$$
\begin{aligned}
P.S. \;=\; & \int \frac{d^4q_1}{(2\pi)^4}(2\pi)^4 \int \frac{d^4q_2}{(2\pi)^4}(2\pi)^4 \delta^{(4)}(p_a + p_b - q_1 - q_2) \\
& \int \frac{d^4p_1}{(2\pi)^4}(2\pi)\delta(p_1^2 - m_1^2)\theta(E_1) \\
& \int \frac{d^4p_2}{(2\pi)^4}(2\pi)\delta(p_2^2 - m_2^2)\theta(E_2) \\
& \int \frac{d^4p_3}{(2\pi)^4}(2\pi)\delta(p_3^2 - m_3^2)\theta(E_3)\delta^{(4)}(q_1 - p_1 - p_2 - p_3) \\
& \int \frac{d^4p_4}{(2\pi)^4}(2\pi)\delta(p_4^2 - m_4^2)\theta(E_4) \\
& \int \frac{d^4p_5}{(2\pi)^4}(2\pi)\delta(p_5^2 - m_5^2)\theta(E_5) \\
& \int \frac{d^4p_6}{(2\pi)^4}(2\pi)\delta(p_6^2 - m_6^2)\theta(E_6)\delta^{(4)}(q_2 - p_4 - p_5 - p_6) \qquad (16)
\end{aligned}
$$

This can be separated in the phase space integral of a $2 \to 2$ reaction $p_a + p_b \to q_1 + q_2$, followed by two decays $q_1 \to p_1 + p_2 + p_3$ and $q_2 \to p_4 + p_5 + p_6$. The only things we miss are the delta functions that fix the masses of q_1 and q_2. This means that we have

$$
\int \frac{d^4q_1}{(2\pi)^4} \;=\; \int \frac{1}{(2\pi)^3} \frac{d^3q_1}{2\omega_1} \frac{dq_1^2}{2\pi} \qquad (17)
$$

in which ω_1 is the energy of the particle and q_1^2 its invariant mass squared. Hence for extras we have to integrate over the invariant mass of the intermediate particle. This makes sense because it is a virtual particle.

From the above it should be clear that we can do the two particles phase space as before with a call to mgoto2. The two decays we can do with calls to the similar routine mgoto3.

Next we can make one important simplification. It is not needed but it does save some work (which you can do for homework). The propagator of an unstable particle is given by $1/((q^2 - m^2) + im\Gamma)$ in which Γ is the decay width. In the matrix element squared this becomes $1/((q^2 - m^2)^2 + m^2\Gamma^2)$. In the narrow width approximation we use

$$
\int \frac{dq^2}{(q^2 - m^2)^2 + m^2\Gamma^2} \;=\; \frac{2\pi}{2m\Gamma} \int dq^2 \delta(q^2 - m^2) \qquad (18)
$$

In other words, we can take the particle on shell and the factor in front works out quite nicely. The $1/(2m)$ is the flux factor for the decay of a particle with a mass m. The decay $q_1 \to p_1 + p_2 + p_3$ would give the partial width for this channel and hence, when divided by the total width we obtain the branching fraction. Hence what we obtain is a complete factorization for production and decay. The only thing that is not factored is the matrix element squared and as a result we retain the so-called spin-spin correllation. We will see what this means.

6

In reaction 4 we will use indeed this kinematical configuration with the narrow width approximation. the files are

- make6

- main6.c

- feett.c

- meett.c

Homework (optional)(very difficult): Modify reaction 4 by inserting the integrals over the mass squared of the intermediate tau's. The width of the tau is given in the file ex4main.c, and so are all other parameters. The width is rather narrow. Hence there is a very sharp spike in the integration area. Very sharp. The chances that VEGAS will find both spikes are super small. Try this and see what happens. Next one can try to improve the two mass integrations with two mappings. These mappings have to compensate a behaviour like $1/((q^2-m^2)^2+m^2\Gamma^2)$. Hint: think about the arctan. If you run into numerical problems, just take the width temporarily a bit bigger for testing purposes. Of course then one has to increase the value of the coupling constant G_F, which is the Fermi constant, as well. Just realize that the width of the τ depends on the square of the Fermi constant. By varying the width you can see that the results are very insensitive to it.

Homework (optional): For the next example we look at the reaction $e^- e^+ \to \gamma\gamma\gamma$. The matrix element resides in the file meeggg.c. The main routine sits in main7.c. Please construct the rest of the program. You are to use the same angle cut as before ($\sin\theta > 0.05$ for all photons) and in addition we assume that each photon has an energy of at least 0.1 GeV. What happens if we don't set that last requirement? Do you have any idea why? And would you know what to do about it?

Finally we go back to our original reaction $e^- e^+ \to \mu^- \mu^+$, but now with also the Z particle in the s-channel. This presents more problems because now we have to introduce the constants in the coupling of the Z to leptons. The coupling to the electron is given by $\gamma^\mu(c_V - c_A\gamma^5)$. Although the coupling to the muon is identical we will indicate it by different symbols in the program. This allows the changing of the program in such a way that we have quarks in the final state. Of course, when there are quarks there are different electrical charges and this will need to be adapted in the matrixelement.

We compute the various c_V and c_A in the main program and pass them on to the matrix element in the form of external variables. In the main program we have to provided a few extra constants: M_Z, M_W and Γ_Z.

The files that are extra for this reaction are main3z.c, feemmz.c, meemmz.c and make3z. For the remainder everything is similar to running the reaction without the Z. Setting the energy is done in the main program as well and it is nice playing to see what the crossections are with and without the Z for a variety of energies.