Integration of three body phase space: The `3BodyXSections` and `3BodyDecays` packages

J. C. Romão*
Departamento de Física and CFTP, Instituto Superior Técnico
Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal

In this note we review the integration of three body phase space. We explain in detail the necessary steps to use two packages for the three body phase space for cross sections as well as for decays. These packages, `3BodyXSections` and `3BodyDecays`, with real examples, can be obtained from my web page [1].

I. INTRODUCTION

It is always a problem when the students want to attack a more complex problem that needs numerical integrations, like the three body decays. In normal textbooks this part it is normally not covered and, at most, is left as an exercise. My experience is that, for the first time, the students need some help. I will try to cover in this article the necessary steps to establish the formulas to be used and then explain how to use two codes in real processes.

II. THE CROSS SECTION

As we want to be as general as possible, we consider the process,

$$p_1(m_a) + p_2(m_b) \to q_1(m_1) + q_2(m_2) + q_3(m_3)\,, \tag{1}$$

without specifying which particles we are referring to. We use for the momenta the conventions of `QGRAF`[2] and `FeynMaster`[3, 4]. For the masses, as we will be mostly interested in the final state, we use m_i to go with q_i. We choose the kinematics in the CM frame shown in Fig 1. The cross section is then given by,

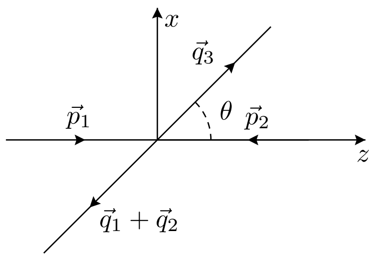

FIG. 1: Kinematics in the CM.

$$d\sigma = \frac{1}{(2\pi)^5}\frac{1}{4\sqrt{(p_1\cdot p_2)^2 - m_a^2 m_b^2}}\;\overline{|\mathcal{M}|^2}\;\delta^4(p_1+p_2-q_1-q_2-q_3)\,\frac{d^3q_1}{2q_1^0}\frac{d^3q_2}{2q_2^0}\frac{d^3q_3}{2q_3^0} \tag{2}$$

*Electronic address: `jorge.romao@ist.utl.pt`

Singling out particle with momenta q_3 and scattering angle, θ, and knowing that the scattering is in one plane, we can do the φ integration to obtain,

$$\sigma = \frac{1}{(2\pi)^5}\frac{1}{4\sqrt{p_1\cdot p_2)^2-m_a^2m_b^2}}\,\frac{2\pi}{2}\int_0^\pi \sin\theta d\theta\int_{m_3}^{E_3^{\max}} dE_3\sqrt{E_3^2-m_3^2}\int\frac{d^3q_1}{2q_1^0}\frac{d^3q_2}{2q_2^0}\,\delta^4(p_1+p_2-p_H-q_1-q_2)\,\overline{|\mathcal{M}|^2}. \quad (3)$$

To proceed we must do the integrations over the the system of particles 1 and 2. For this we evaluate the following Lorentz invariant quantity

$$X = \int\frac{d^3q_1}{2q_1^0}\frac{d^3q_2}{2q_2^0}\,\delta^4(\Delta-q_1-q_2)\,\overline{|\mathcal{M}|^2}\,. \quad (4)$$

where $\Delta = p_1+p_2-q_3$. These quantities are Lorentz invariant and can be evaluated in any reference frame. However the calculation is particularly simple in the CM frame of particles 1 and 2. In that frame,

$$\Delta = (m_{12},0,0,0)\,, \quad (5)$$

where

$$m_{12}^2 = (q_1+q_2)^2 = (p_1+p_2-q_3)^2 = s+m_3^2-2\sqrt{s}E_3, \quad (6)$$

is the invariant mass of the pair. We then get (q_i^* are the vectors in the CM frame of the pair)

$$\begin{aligned} X =&\frac{1}{4}\int\frac{d^3q_1^*}{q_1^{*0}q_2^{*0}}\,\delta(m_{12}-q_1^{*0}-q_2^{*0})\overline{|\mathcal{M}|^2}\\ =&\frac{1}{4}\int\frac{|\vec{q}_1^{\,*}|^2 d|\vec{q}_1^{\,*}|\,d\Omega_1^*}{q_1^{*0}q_2^{*0}}\,\delta(m_{12}-q_1^{*0}-q_2^{*0})\overline{|\mathcal{M}|^2} \end{aligned} \quad (7)$$

But in this CM frame we have $|\vec{q}_1^{\,*}| = |\vec{q}_2^{\,*}|$ and therefore

$$q_1^{*0} = \sqrt{|\vec{q}_1^{\,*}|^2+m_1^2}, \quad q_2^{*0} = \sqrt{|\vec{q}_1^{\,*}|^2+m_2^2}. \quad (8)$$

We get then

$$\begin{aligned} X =&\frac{1}{4}\int\frac{|\vec{q}_1^{\,*}|^2 dq_1^{*0}\,d\Omega_1^*}{q_1^{*0}q_2^{*0}}\,\delta\left(m_{12}-\sqrt{|\vec{q}_1^{\,*}|^2+m_1^2}-\sqrt{|\vec{q}_1^{\,*}|^2+m_2^2}\right)\overline{|\mathcal{M}|^2}\\ =&\frac{1}{4}\int d\Omega_1^*\,\frac{|\vec{q}_1^{\,*}|^2}{q_1^{*0}q_2^{*0}}\,\frac{1}{|\vec{q}*_1|\dfrac{q_1^{*0}+q_2^{*0}}{q_1^{*0}q_2^{*0}}}\overline{|\mathcal{M}|^2}\\ =&\frac{1}{4}\frac{|\vec{q}_1^{\,*}|}{m_{12}}\int d\Omega_1^*\,\overline{|\mathcal{M}|^2}\\ =&\frac{1}{8}\frac{\lambda(m_{12},m_1,m_2)}{m_{12}^2}\int d\Omega_1^*\,\overline{|\mathcal{M}|^2} \end{aligned} \quad (9)$$

where we have used

$$|\vec{q}_1^{\,*}| = \frac{\lambda(m_{12},m_1,m_2)}{2m_{12}} \quad (10)$$

with λ being the Källen function,

$$\lambda(x,y,z) = \sqrt{x^4+y^4+z^4-2x^2y^2-2x^2z^2-2y^2z^2}. \quad (11)$$

Notice that Eq. (9) is symmetric in the interchange of particles 1 and 2, indicating that the order of integration in $\vec{q}_1^{\,*}$ or $\vec{q}_2^{\,*}$ is arbitrary. The final form for the cross section is then,

$$\sigma = \frac{1}{64(2\pi)^4}\frac{1}{\sqrt{p_1\cdot p_2)^2-m_a^2m_b^2}}\int_{m_3}^{E_3^{\max}} dE_3\sqrt{E_3^2-m_3^2}\int_0^\pi d\theta\sin\theta\int_0^\pi d\theta^*\sin\theta^*\int_0^{2\pi} d\varphi^*\,\frac{\lambda(m_{12},m_1,m_2)}{m_{12}^2}\,\overline{|\mathcal{M}|^2} \quad (12)$$

and we have already used the fact that the final result does not depend on the azimuthal angle φ of particle 3 and have done that integration. Note that if we have identical particles (see example below) we have still to divide by the symmetry factor in the end of the calculation. It is crucial to understand the meaning of the angles in the previous expression. The angle θ is the scattering angle of the Higgs boson in the CM of the collision (for collider also the lab frame), while the angles (θ^*, φ^*) are the angles of one of the two particles (the one with momentum q_1) in the CM frame of the pair. It is in these variables that the integrations are simple to set, but of course, there remains the difficulty of relating the 4-momenta in that frame to the lab frame. We will discuss in generality this question in the following section.

III. KINEMATICS: HOW TO RELATE DIFFERENT REFERENCE FRAMES

As we have seen in the previous section we have to relate the coordinates of 4-vectors in different reference frames. The best way is to do this in steps, using in sequence elementary rotations and boosts. We always perform boosts along the z axis, so the relevant Lorentz transformation is

$$x = \mathbf{Boost_z}(\beta) \cdot x' \tag{13}$$

where $x(x')$ are the coordinates in the reference frame $S(S')$, respectively, and the reference frame S' moves with relative speed β along the z axis with respect to S. The explicit form of the matrix $\mathbf{Boost_z}(\beta)$ is

$$\mathbf{Boost_z}(\beta) = \begin{bmatrix} \gamma & 0 & 0 & \gamma\beta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \gamma\beta & 0 & 0 & \gamma \end{bmatrix} \tag{14}$$

For rotations, we can always use a sequence of rotations along the coordinate axis. So the relevant rotations, always written with the convention

$$x = \mathbf{Rot_x}(\theta) \cdot x' \quad ; \quad x = \mathbf{Rot_y}(\theta) \cdot x' \quad ; \quad x = \mathbf{Rot_z}(\theta) \cdot x' \tag{15}$$

are

$$\mathbf{Rot_x}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad \mathbf{Rot_y}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & \sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{16}$$

and

$$\mathbf{Rot_z}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

Now, if we look at the kinematics shown in Fig. 1, we can easily realize that the correct transformation laws for q_1^* and q_2^* in the CM frame of the Z boson pair are,

$$q_{1,2} = \mathbf{Rot_y}(\theta + \pi) \cdot \mathbf{Boost_z}(\beta) \cdot q_{1,2}^* \tag{18}$$

where

$$\beta = \sqrt{1 - \frac{1}{\gamma^2}}, \quad \gamma = \frac{E_{12}}{m_{12}}, \quad E_{12} = \frac{s + m_{12}^2 - m_3^2}{2\sqrt{s}},$$

$$m_{12} = \sqrt{(q_1 + q_2)^2} = \sqrt{s + m_3^2 - 2\sqrt{s}E_3}. \tag{19}$$

Taking in account that

$$\begin{cases} q_1^{*0} = \dfrac{m_{12}^2 + m_1^2 - m_2^2}{2m_{12}} \\ q_1^{*1} = q_{12}^{\text{CM}} \sin\theta^* \cos\varphi^* \\ q_1^{*2} = q_{12}^{\text{CM}} \sin\theta^* \sin\varphi^* \\ q_1^{*3} = q_{12}^{\text{CM}} \cos\theta^* \end{cases} \quad \begin{cases} q_2^{*0} = \dfrac{m_{12}^2 + m_2^2 - m_1^2}{2m_{12}} \\ q_2^{*1} = -q_{12}^{\text{CM}} \sin\theta^* \cos\varphi^* \\ q_2^{*2} = -q_{12}^{\text{CM}} \sin\theta^* \sin\varphi^* \\ q_2^{*3} = -q_{12}^{\text{CM}} \cos\theta^* \end{cases} \tag{20}$$

where

$$q_{12}^{\rm CM} = \frac{\lambda(m_{12}, m_1, m_2)}{2m_{12}}, \tag{21}$$

we can now use Eq. (18) to obtain q_1 and q_2 in the lab frame, as a function of $(E_3, \theta, \theta^*, \varphi^*)$.

There is still one important point in the kinematics, that is to know what are the integration limits on the energy E_3 of particle 3 in Eq. (12). Obviously the minimum of the energy is for $E_3 = m_3$. For the maximum value, it can be easily obtained, as we know that the momentum of particle 3 is given by

$$|\vec{q}_3| = \frac{\lambda(\sqrt{s}, m_{12}, m_3)}{2\sqrt{s}}. \tag{22}$$

Now the maximum value for E_3 is obtained for the maximum value of $|\vec{q}_3|$. This occurs for[5],

$$m_{12} = m_1 + m_2\,. \tag{23}$$

So we get,

$$|\vec{q}_3|^{\rm max} = \frac{\lambda(\sqrt{s}, m_1 + m_2, m_3)}{2\sqrt{s}}, \quad E_3^{\rm max} = \sqrt{(|\vec{q}_3|^{\rm max})^2 + m_3^2}\,. \tag{24}$$

IV. DECAY WIDTH

After explaining in detail the three body phase space for the CM cross section, we can easily obtain the expression for the three body decay of one particle. For definiteness we consider the process,

$$P(M) \to q_1(m_1) + q_2(m_2) + q_3(m_3) \tag{25}$$

where for the final state we use the same conventions as before. Now the formula for the decay width can be easily obtained from that for the cross section, Eq. (12), by the substitution,

$$4\sqrt{(p_1 \cdot p_2)^2 - m_a^2 m_b^2} \to 2M\,. \tag{26}$$

We get therefore[1]

$$\Gamma = \frac{1}{16(2\pi)^4}\,\frac{1}{2M}\int_{m_3}^{E_3^{\rm max}} dE_3 \sqrt{E_3^2 - m_3^2}\int_0^{\pi} d\theta \sin\theta \int_0^{\pi} d\theta^* \sin\theta^* \int_0^{2\pi} d\varphi^* \frac{\lambda(m_{12}, m_1, m_2)}{m_{12}^2}\,\overline{|\mathcal{M}|^2}, \tag{27}$$

with the same integration over the three body phase space.

V. NUMERICAL INTEGRATION PACKAGES

With the kinematics completely implemented, we have to perform the integrations and obtain the cross sections. In few problems can the integrals that appear in the evaluation of cross sections be done analytically. Most of the time we have to revert to numerical methods to evaluate the integrals. There are many ways to do evaluate the integrals numerically. A good library is `CUBA package`[6]. It can be linked either with `C/C++` or with `Fortran` programs. It has several methods with the same calling structure, so that one can substitute one method for another and check for accuracy. The manual comes with the package. I have done a Gaussian integration subroutine in `Fortran` called `gauss.f` that uses the same calling conventions, and therefore can also be tested in the same way. Although the method does not differ for cross section and decays, it is useful to have to different ready made programs for the two cases. These are described in the next subsections and can be downloaded from my web page[1].

---

[1] One might wonder about the dimensions in view of Eq. (26). But we have to realize that the dimensions of $\mathcal{M}$ are M^0 for the process in Eq. (25) and M^{-1} for the process in Eq. (1). This gives $[\sigma] = M^{-2}$ while $[\Gamma] = M$ as it should.

A. Package 3BodyXSections

The package `3BodyXSections` has three main components, that we will briefly explain below

- `Driver3BodyXS.f`

  This is the driver program that call the integration routine. It is in this program that the constants are defined. It has a simple structure,

```
      program XS3Body
      implicit none
      integer i,method,npoints

      real*8 rs,ma,mb,m1,m2,m3
      real*8 thetaw,g2,gp,mw,mz,gw,gz,sw2,ee,GF,pi
      real*8 xs(1),xsection

      common/data/mw,mz,gw,gz,sw2,ee,GF,pi
      common/parameters/rs,ma,mb,m1,m2,m3

      open(10,file='res.dat',status='unknown')

      ...

      call sigma(xs,method,npoints)

      ...

      end
```

  In this program `method` refers to the integration method, `method=1` for Gaussian integration and `method=2` for Vegas integration. Vegas is normally better and faster for the four dimensional integrations that we have to perform. Gauss integration is included for comparison. If you choose Gaussian integration you have to specify the number of `points`, see inside the integration subroutine for more details.

- `Integration3Body.f`

  This file includes the integration routines and gives the value of the cross section. In principle the user does not have to change this file. The important points are,

```
      subroutine sigma(xs,method,npoints)
      implicit none

      ...

      xs(1)=integral(1)

      return
      end
*----------------------------------------------------------------------*
*----------------------------------------------------------------------*
      subroutine integrand(ndim, xx, ncomp, ff)
      implicit none

       ...

      ff(1) = dsig(x)*Jacob

      return
      end
```

```
*----------------------------------------------------------------------*
*----------------------------------------------------------------------*
      real*8 function dsig(x)
      implicit none
      real*8 theta,theta1
      real*8 rs,ma,mb,m1,m2,m3
      real*8 p1p2,p1q1,p1q2,p1q3,p2q1,p2q2,p2q3,q1q2,q1q3,q2q3,
     &     m12,p1p1,p2p2,q1q1,q2q2,q3q3,EPSp1p2q1q2

      ...

      dsig=cst/Flux*sin(theta)*sin(theta1)*2d0*pi/(2d0*pi)**5*p3CM*
     &     Msq(rs,ma,mb,m1,m2,m3,p1p2,p1q1,p1q2,
     &     p1q3,p2q1,p2q2,p2q3,q1q2,q1q3,q2q3,m12,p1p1,p2p2,q1q1,
     &     q2q2,q3q3,EPSp1p2q1q2)


      return
      end
*----------------------------------------------------------------------*
*----------------------------------------------------------------------*
      subroutine TBkinXS(x,rs,ma,mb,m1,m2,m3,p1p2,p1q1,p1q2,
     &     p1q3,p2q1,p2q2,p2q3,q1q2,q1q3,q2q3,m12,p1p1,p2p2,q1q1,
     &     q2q2,q3q3,EPSp1p2q1q2)
      implicit none

      ...

      q1q2=dot(q1,q2)
      q1q3=dot(q1,q3)
      q2q3=dot(q2,q3)

      EPSp1p2q1q2=epsilon(p1,p2,q1,q2)

      return
      end
*----------------------------------------------------------------------*
```

The subroutine `TBkinXS` gives, for each point $(E_3, \theta, \theta^*, \varphi^*)$, all the Lorentz invariant scalar products. The only point worth mention is that the user has to supply the averaged squared amplitude as described below. We have also included here the subroutines necessary for the integration of three body decays, see below. So you need only one file `Integration3Body.f` for both cases.

- `MsqAvg.f`

  This file contains the user supplied averaged squared amplitude for the case in study. The syntax of the calling function it is important

```
*----------------------------------------------------------------------*
        real*8 function Msq(rs,ma,mb,m1,m2,m3,p1p2,p1q1,p1q2,
     &     p1q3,p2q1,p2q2,p2q3,q1q2,q1q3,q2q3,m12,p1p1,p2p2,q1q1,
     &     q2q2,q3q3,EPSp1p2q1q2)
      implicit none
      real*8 rs,ma,mb,m1,m2,m3,p1p2,p1q1,p1q2,
     &     p1q3,p2q1,p2q2,p2q3,q1q2,q1q3,q2q3,m12,p1p1,p2p2,q1q1,
     &     q2q2,q3q3,EPSp1p2q1q2

       ...
```

```
      Msq=...
      end
*----------------------------------------------------------------------*
```

The arguments of this function are all the scalar products, written in obvious notation

$$\mathrm{p1q1} = (p_1 \cdot q_1), \mathrm{p1q2} = (p_1 \cdot q_2), \ldots \tag{28}$$

From my web site [1], you can download the package `3BodyXSections.tar.gz` that contain all the files for the process $e^- e^+ \to ZZH$.

B. Package `3BodyDecays`

The package `3BodyDecays` has again the corresponding three main components, that we will review below:

- `Driver3BodyDecays.f`

  This file is the driver program for the three body decay.

```
*----------------------------------------------------------------------*
      program Decays3Body
      implicit none
      integer i,method,npoints

      ...

      call DecayWidth(width,method,npoints)

      ...

      end
*----------------------------------------------------------------------*
```

- `Integration3Body.f`

  This file incorporates the integration routines. It is the same file for three body decays and three body cross sections, all the necessary subroutines are included. In principle, the user does not have to change this file.

```
*----------------------------------------------------------------------*
      subroutine DecayWidth(width,method,npoints)
      implicit none
      ...
      end
*----------------------------------------------------------------------*
      subroutine integrand(ndim, xx, ncomp, ff)
      implicit none

      ff(1) = dgamma(x)*Jacob

      end
*----------------------------------------------------------------------*
      real*8 function dgamma(x)
      implicit none
      ...
      call TBkinDecays(x,M0,m1,m2,m3,P0q1,P0q2,P0q3,q1q2,q1q3,q2q3,
     &     m12,P0P0,q1q1,q2q2,q3q3,EPSP0q1q2q3)
      ...
      end
```

```
*-----------------------------------------------------------------------*
      subroutine TBkinDecays(x,M0,m1,m2,m3,P0q1,P0q2,P0q3,q1q2,q1q3,
     &     q2q3,m12,P0P0,q1q1,q2q2,q3q3,EPSP0q1q2q3)
      implicit none

      ...

      end
*-----------------------------------------------------------------------*
```

- `MsqAvg.f`

  This file evaluates the averaged squared amplitude, like in the previous case,

```
*-----------------------------------------------------------------------*
      real*8 function Msq(M0,m1,m2,m3,P0q1,P0q2,P0q3,q1q2,q1q3,q2q3,
     &     m12,P0P0,q1q1,q2q2,q3q3,EPSP0q1q2q3)
      implicit none
      ...

      MSq=...
      end
*-----------------------------------------------------------------------*
```

  where `Msq` is supplied by the user.

VI. EXAMPLES

In this section we give examples of the use for these packages.

A. The cross section for $e^- + e^+ \to Z + Z + H$

As a first example we look at the process

$$e^-(p_1) + e^+(p_2) \to Z(q_1) + Z(q_2) + H(q_3). \tag{29}$$

As explained in my text in Ref.[7], we have four diagrams and the averaged squared amplitude can be written, in an obvious notation, as

$$\overline{|\mathcal{M}|^2} = \frac{1}{4}\left[|\mathcal{M}_{11}|^2 + |\mathcal{M}_{22}|^2 + |\mathcal{M}_{33}|^2 + |\mathcal{M}_{44}|^2 + |\mathcal{M}_{12}|^2 + |\mathcal{M}_{13}|^2 + |\mathcal{M}_{14}|^2 + |\mathcal{M}_{23}|^2 + |\mathcal{M}_{24}|^2 + |\mathcal{M}_{34}|^2\right] \tag{30}$$

We reproduce below the code for the file `MsqAvg.f`. In the package `3BodyXSections` this is the example included.

```
*-----------------------------------------------------------------------*
      real*8 function Msq(rs,ma,mb,m1,m2,m3,p1p2,p1q1,p1q2,
     &     p1q3,p2q1,p2q2,p2q3,q1q2,q1q3,q2q3,m12,p1p1,p2p2,q1q1,
     &     q2q2,q3q3,EPSp1p2q1q2)
      implicit none
      real*8 rs,ma,mb,m1,m2,m3,p1p2,p1q1,p1q2,
     &     p1q3,p2q1,p2q2,p2q3,q1q2,q1q3,q2q3,m12,p1p1,p2p2,q1q1,
     &     q2q2,q3q3,EPSp1p2q1q2
      real*8 gV,gA,MH,cst
      real*8 Msq11,Msq22,Msq33,Msq44,Msq12,Msq13,Msq14
      real*8 Msq23,Msq24,Msq34
      real*8 mw,mz,gw,gz,sw2,ee,GF,pi
      real*8 p3CMmax,E3max,kallen
```

```
 common/data/mw,mz,gw,gz,sw2,ee,GF,pi
 ga=-1d0/4d0
 gv=-1d0/4d0+sw2
 MH=m3


   Msq11=(-2*(ga**4 + 6*ga**2*gv**2 + gv**4)*(-16*p1q1*p2q1
&  *p2q2**2 + MZ**6*rs**2 -4*MZ**2*p2q2*(-4*p1q1*p2q1 + 4*p1q1
&  *q1q2 +p2q2*rs**2) +4*MZ**4*(p1q1*p2q1 + p2q2*(-2*p1q2
&  + rs**2))))/(MZ**4*(MZ**2 - 2*p2q2)**2*(MH**4 + GZ**2*MZ**2 + 4
&  *MH**2*q1q3 + 4*q1q3**2))

   Msq22=(-2*(ga**4 + 6*ga**2*gv**2 + gv**4)*(-16*p1q1**2*p1q2
&  *p2q2 + MZ**6*rs**2 -4*MZ**2*p1q1*(-4*p1q2*p2q2 + 4*p2q2*q1q2
&  +p1q1*rs**2) +MZ**4*(-8*p1q1*p2q1 + 4*p1q2*p2q2 + 4*p1q1
&  *rs**2)))/(MZ**4*(MZ**2 - 2*p1q1)**2*(MH**4 + GZ**2*MZ**2 + 4
&  *MH**2*q2q3 + 4*q2q3**2))

   Msq33=(-2*(ga**4 + 6*ga**2*gv**2 + gv**4)*(-16*p1q1*p1q2**2
&  *p2q1 + MZ**6*rs**2 -4*MZ**2*p1q2*(-4*p1q1*p2q1 + 4*p2q1*q1q2
&  +p1q2*rs**2) +4*MZ**4*(p1q1*p2q1 + p1q2*(-2*p2q2 + rs**2))))
&  /(MZ**4*(MZ**2 - 2*p1q2)**2*(MH**4 + GZ**2*MZ**2 + 4*MH**2
&  *q1q3 + 4*q1q3**2))

   Msq44=(-2*(ga**4 + 6*ga**2*gv**2 + gv**4)*(-16*p1q2*p2q1**2
&  *p2q2 + MZ**6*rs**2 -4*MZ**2*p2q1*(-4*p1q2*p2q2 + 4*p1q2*q1q2
&  +p2q1*rs**2) +MZ**4*(-8*p1q1*p2q1 + 4*p1q2*p2q2 + 4*p2q1
&  *rs**2)))/(MZ**4*(MZ**2 - 2*p2q1)**2*(MH**4 + GZ**2*MZ**2 + 4
&  *MH**2*q2q3 + 4*q2q3**2))

   Msq12=(64*EPSp1p2q1q2*ga*gv*(ga**2 + gv**2)*GZ*(MZ**2 + 2*p1q1)
&  *(MZ**2 + 2*p2q2)*(q1q3 - q2q3))/(MZ**3*(MZ**2 - 2*p1q1)
&  *(MZ**2 - 2*p2q2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q1q3 + 4
&  *q1q3**2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q2q3 + 4*q2q3**2)) -(4
&  *(ga**4 + 6*ga**2*gv**2 + gv**4)*(MH**4 + GZ**2*MZ**2 + 4*q1q3
&  *q2q3 +2*MH**2*(q1q3 + q2q3))*(4*p1q1*p2q2*(2*p1q2*p2q1 + 2
&  *p1q1*p2q2 -q1q2*rs**2) +MZ**4*(2*p1q2*p2q1 + 18*p1q1*p2q2
&  - q1q2*rs**2) -2*MZ**2*(p1q1 + p2q2)*(-2*p1q2*p2q1 + 6*p1q1
&  *p2q2 + q1q2*rs**2)))/(MZ**4*(MZ**2 - 2*p1q1)*(MZ**2 - 2*p2q2)
&  *(MH**4 + GZ**2*MZ**2 + 4*MH**2*q1q3 + 4*q1q3**2)*(MH**4
&  + GZ**2*MZ**2 + 4*MH**2*q2q3 + 4*q2q3**2))


   Msq13=(4*(ga**4 + 6*ga**2*gv**2 + gv**4)*(-16*p1q1*p1q2*p2q1
&  *p2q2 + 5*MZ**6*rs**2 +MZ**2*(8*p1q2*p2q1**2 + 8*p1q1**2*p2q2
&  +8*p1q1*p2q1*rs**2 - 4*p1q2*p2q2*rs**2 -4*p1q1*q1q2*rs**2 - 4
&  *p2q1*q1q2*rs**2) -2*MZ**4*(2*p1q1*p2q1 + rs**2*(p1q2 + p2q2
&  - rs**2))))/(MZ**4*(MZ**2 - 2*p1q2)*(MZ**2 - 2*p2q2)*(MH**4
&  + GZ**2*MZ**2 + 4*MH**2*q1q3 + 4*q1q3**2))


   Msq14=(64*EPSp1p2q1q2*ga*gv*(ga**2 + gv**2)*GZ*(3*MZ**4 - 4
&  *p2q1*p2q2 - 2*MZ**2*(p2q1 + p2q2))*(q1q3 - q2q3))/(MZ**3
&  *(MZ**2 - 2*p2q1)*(MZ**2 - 2*p2q2)*(MH**4 + GZ**2*MZ**2 + 4
&  *MH**2*q1q3 + 4*q1q3**2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q2q3
&  + 4*q2q3**2)) +(4*(ga**4 + 6*ga**2*gv**2 + gv**4)*(MH**4
&  + GZ**2*MZ**2 + 4*q1q3*q2q3 +2*MH**2*(q1q3 + q2q3))*(MZ**4*(2
```

```
     &  *p1q2*p2q1 + 2*p1q1*p2q2 +(8*p2q1 + 8*p2q2 - 5*q1q2)*rs**2) +4
     &  *p2q1*p2q2*(2*p1q2*p2q1 + 2*p1q1*p2q2 -q1q2*rs**2) +2*MZ**2*(2
     &  *p1q2*p2q1*(p2q1 - 3*p2q2) +2*p1q1*p2q2*(-3*p2q1 + p2q2)
     &  -(p2q1 + p2q2)*q1q2*rs**2)))/(MZ**4*(MZ**2 - 2*p2q1)*(MZ**2 - 2
     &  *p2q2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q1q3 + 4*q1q3**2)*(MH**4
     &  + GZ**2*MZ**2 + 4*MH**2*q2q3 + 4*q2q3**2))

        Msq23=(64*EPSp1p2q1q2*ga*gv*(ga**2 + gv**2)*GZ*(3*MZ**4 - 4
     &  *p1q1*p1q2 - 2*MZ**2*(p1q1 + p1q2))*(q1q3 - q2q3))/(MZ**3
     &  *(MZ**2 - 2*p1q1)*(MZ**2 - 2*p1q2)*(MH**4 + GZ**2*MZ**2 + 4
     &  *MH**2*q1q3 + 4*q1q3**2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q2q3
     &  + 4*q2q3**2)) +(4*(ga**4 + 6*ga**2*gv**2 + gv**4)*(MH**4
     &  + GZ**2*MZ**2 + 4*q1q3*q2q3 +2*MH**2*(q1q3 + q2q3))*(4*p1q1
     &  *p1q2*(2*p1q2*p2q1 + 2*p1q1*p2q2 -q1q2*rs**2) +MZ**4*(2*p1q1
     &  *p2q2 + 8*p1q1*rs**2 - 5*q1q2*rs**2 +2*p1q2*(p2q1 + 4*rs**2))
     &  +2*MZ**2*(2*p1q1**2*p2q2 +p1q2*(2*p1q2*p2q1 - q1q2*rs**2) -p1q1
     &  *(6*p1q2*(p2q1 + p2q2) + q1q2*rs**2))))/(MZ**4*(MZ**2 - 2*p1q1)
     &  *(MZ**2 - 2*p1q2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q1q3 + 4
     &  *q1q3**2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q2q3 + 4*q2q3**2))


        Msq24=(4*(ga**4 + 6*ga**2*gv**2 + gv**4)*(-16*p1q1*p1q2*p2q1
     &  *p2q2 + 5*MZ**6*rs**2 -4*MZ**2*(-2*p1q2**2*p2q1 - 2*p1q1
     &  *p2q2**2 +p1q1*p2q1*rs**2 + p2q2*q1q2*rs**2 +p1q2*(-2*p2q2
     &  + q1q2)*rs**2) -2*MZ**4*(2*p1q2*p2q2 + rs**2*(p1q1 + p2q1
     &  - rs**2))))/(MZ**4*(MZ**2 - 2*p1q1)*(MZ**2 - 2*p2q1)*(MH**4
     &  + GZ**2*MZ**2 + 4*MH**2*q2q3 + 4*q2q3**2))

        Msq34=(64*EPSp1p2q1q2*ga*gv*(ga**2 + gv**2)*GZ*(MZ**2 + 2*p1q2)
     &  *(MZ**2 + 2*p2q1)*(q1q3 - q2q3))/(MZ**3*(MZ**2 - 2*p1q2)
     &  *(MZ**2 - 2*p2q1)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q1q3 + 4
     &  *q1q3**2)*(MH**4 + GZ**2*MZ**2 + 4*MH**2*q2q3 + 4*q2q3**2)) -(4
     &  *(ga**4 + 6*ga**2*gv**2 + gv**4)*(MH**4 + GZ**2*MZ**2 + 4*q1q3
     &  *q2q3 +2*MH**2*(q1q3 + q2q3))*(4*p1q2*p2q1*(2*p1q2*p2q1 + 2
     &  *p1q1*p2q2 -q1q2*rs**2) +MZ**4*(18*p1q2*p2q1 + 2*p1q1*p2q2
     &  - q1q2*rs**2) -2*MZ**2*(p1q2 + p2q1)*(6*p1q2*p2q1 - 2*p1q1
     &  *p2q2 + q1q2*rs**2)))/(MZ**4*(MZ**2 - 2*p1q2)*(MZ**2 - 2*p2q1)
     &  *(MH**4 + GZ**2*MZ**2 + 4*MH**2*q1q3 + 4*q1q3**2)*(MH**4
     &  + GZ**2*MZ**2 + 4*MH**2*q2q3 + 4*q2q3**2))


        Msq=1d0/4d0*(Msq11+Msq22+Msq33+Msq44+Msq12+Msq13
     &       +Msq14+Msq23+Msq24+Msq34)

        cst=256d0*GF**3/sqrt(2d0)*MZ**8

        Msq=Msq*cst

        return
        end
*-------------------------------------------------------------------*
```

In Fig. 2 we present a comparison of our result with the CalcHEP result, for the same parameters, showing complete agreement. CalcHEP also uses Vegas integration in its code. At this point one might ask why do not use CalcHEP all the time? The answer has to do with the fact that with our code we do not need to have a model implemented in CalcHEP, and in some cases that is useful. The other reason is that in this way we control the code and can look at other aspects, like for instance the cross section for producing longitudinally polarized Z bosons which would be difficult in CalcHEP.

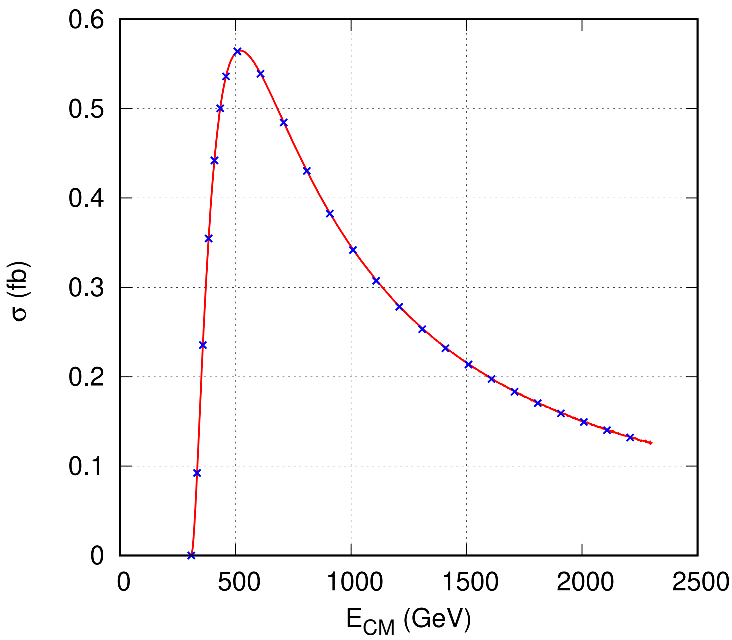

FIG. 2: Cross section for $e^- + e^+ \to Z + Z + H$. The red line corresponds to our calculation, the blue dots comparison with `CalcHEP`

B. The decay $H \to e^- + \overline{\nu}_e + W^+$

As an example of the `3BodyDecays` package we look at the decay $H(P) \to e^-(q_1) + \overline{\nu}_e(q_2) + W^+(q_3)$. Before the Higgs boson was discovered it was common to plot the decay widths, for various channels, as a function of the Higgs boson mass. This decay is particularly interesting as it can happen that the intermediate W boson can be close to on-shell, enhancing the decay width. Neglecting the electron mass the only diagram contributing is shown in Fig. 3. This leads to the following amplitude

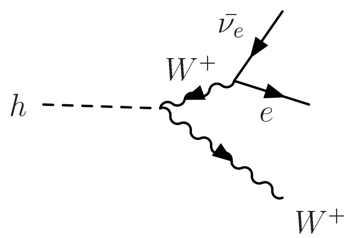

FIG. 3: Diagram for the decay $H \to e^- + \overline{\nu}_e + W^+$.

$$\mathcal{M} = -\frac{g^2}{\sqrt{2}} \frac{1}{(P-q_3)^2 - M_W^2 + i\Gamma_W M_W} \overline{u}(q_1)\gamma^\mu P_L v(q_2)\epsilon_\mu(q_3) \tag{31}$$

A simple calculation leads to the following expression

$$\overline{|\mathcal{M}|^2} = g^4 \left[(q_1 \cdot q_2) + \frac{2(q_1 \cdot q_3)(q_2 \cdot q_3)}{M_W^2} \right] \frac{M_W^2}{(M_H^2 - 2(P \cdot q_3))^2 + M_W^2 \Gamma_W^2} \tag{32}$$

With this we can evaluate the decay width as a function of the Higgs boson mass. The result is shown in Fig. 3 where again we compare with the `CalcHEP` result.

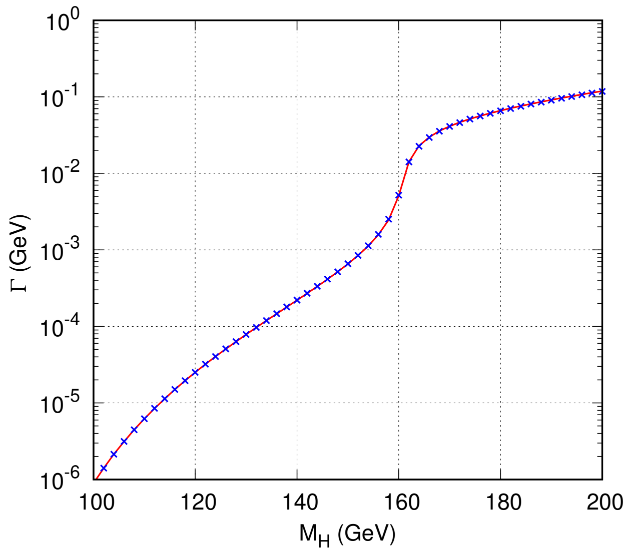

FIG. 4: Decay width $H \to e^- + \overline{\nu}_e + W^+$ as a function of the Higgs boson mass.The red line corresponds to our calculation, the blue dots comparison with `CalcHEP`.

VII. CONCLUSIONS

In this note we review the integration of three body phase space. We explain in detail the necessary steps to use two packages for the three body phase space for cross sections as well as for decays. We show examples of the use of these tools in the calculation of the cross section for $e^- + e^+ \to Z + Z + H$ and the decay width of $H \to e^- + \overline{\nu}_e + W^+$. These packages, `3BodyXSections` and `3BodyDecays`, with real examples, can be obtained from my web page [1].

[1] J. C. Romao, http://porthos.ist.utl.pt/CTQFT/ .
[2] P. Nogueira, http://cfif.ist.utl.pt/ paulo/qgraf.html .
[3] D. Fontes and J. C. Romao, Comput. Phys. Commun. **256**, 107311 (2020), [1909.05876].
[4] D. Fontes and J. C. Romão, JHEP **06**, 016 (2021), [2103.06281].
[5] Particle Data Group, P. A. Zyla *et al.*, PTEP **2020**, 083C01 (2020).
[6] T. Hahn, http://www.feynarts.de/cuba/ .
[7] J. C. Romao, https://porthos.tecnico.ulisboa.pt/CTQFT/files/RealExample.pdf (2022).