

# Neutron star cooling by axion emission

Ray Hagimoto

August 10, 2023

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Integration in terms of momentum variables (didn't work) . . . . .	2
1.2	A recursive expression for phase space integrals . . . . .	3
1.3	Writing the integral in terms of momentum transfers . . . . .	5
<b>2</b>	<b>Monte Carlo integration</b>	<b>7</b>
2.1	The basics . . . . .	7
2.2	Importance sampling . . . . .	8
2.3	Stratified sampling . . . . .	9
2.4	Multichannel sampling . . . . .	9
<b>3</b>	<b>CompHEP and MADGRAPH5</b>	<b>9</b>
3.1	CompHEP . . . . .	9
<b>A</b>	<b>Derivation of recursive relation</b>	<b>10</b>

## 1 Introduction

The goal of this project is to evaluate the integral

### Emissivity integral

$$\varepsilon_{3'} = \int f_1 f_2 (1 - f_{1'}) (1 - f_{2'}) (1 + f_{3'}) \times E_{3'} \sum_{\sigma, \sigma'} |\mathcal{M}|^2 d\Phi_3((p_1 + p_2)^2; 1', 2', 3') \frac{d^3 p_1}{(2\pi)^3 2E_1} \frac{d^3 p_2}{(2\pi)^3 2E_2} \quad (1.1)$$

where,

### Lorentz invariant phase space measure

$$d\Phi_n((p_a + p_b)^2; 1, \dots, n) \equiv \prod_{i=1}^n \left[ \delta(p_i^2 - m_i^2) \Theta(p_i^0) \frac{d^4 p_i}{(2\pi)^3} \right] (2\pi)^4 \delta^4 \left( p_a + p_b - \sum_{i=1}^n p_i \right) \quad (1.2)$$

is the Lorentz-invariant phase space measure, and

$$f_i \equiv f_{FD}(E_i) \equiv \frac{1}{e^{(E_i - \mu_i)/T} + 1}$$

is the Fermi-Dirac distribution. The spin-summed matrix element squared is given by

### Spin-summed matrix element

$$\sum_{\sigma, \sigma'} |\mathcal{M}|^2 = \frac{128 g_{ae\mu}^2 e^4}{(m_1^2 - m_{1'}^2)^2} \frac{(p_1 \cdot p_{1'} - m_1 m_{1'})(p_2 \cdot p_{3'})(p_{2'} \cdot p_{3'})}{(p_2 - p_{2'})^4} . \quad (1.3)$$

### 1.1 Integration in terms of momentum variables (didn't work)

*The first strategy I used to evaluate this integral was to do the integration directly in terms of the momentum variables. Below are the notes I wrote detailing my strategy.*

We can use the momentum conserving Dirac delta to evaluate the  $\mathbf{p}_{2'}$  integral, setting

$$\mathbf{p}_{2'} = \mathbf{p}_1 + \mathbf{p}_2 - \mathbf{p}_{1'} - \mathbf{p}_{3'} . \quad (1.4)$$

We now choose to align the  $z$ -axis with  $\mathbf{p}_1$  and measure angles with respect to this axis. Converting to spherical polar coordinates gives, for example,  $d^3 p_{3'} = p_{3'}^2 dp_{3'} d\cos\theta_{13'} d\phi_{13'}^1$ . Then the energy conserving Dirac delta can be used to evaluate the  $dp_{3'}$  integral in the

<sup>1</sup>The angle  $\phi_{13'}$  is not measured with respect to  $\mathbf{p}_1$ , it is measured to some axis orthogonal to  $\mathbf{p}_1$ . We don't need to define that axis explicitly as long as the other angles  $\phi_{1i}$  are measured with respect to the same axis.

following way. First, we use  $E = \sqrt{p^2 + m^2}$  to rewrite the masses. Then we use momentum conservation to make the replacement  $\mathbf{p}_{2'} \rightarrow \mathbf{p}_1 + \mathbf{p}_2 - \mathbf{p}_{1'} - \mathbf{p}_{3'}$ :

$$\begin{aligned}
E_1 + E_2 &= E_{1'} + E_{2'} + E_{3'} \\
\Rightarrow \sqrt{p_1^2 + m_1^2} + \sqrt{p_2^2 + m_2^2} &= \sqrt{p_{1'}^2 + m_{1'}^2} + \sqrt{p_{2'}^2 + m_{2'}^2} + \sqrt{p_{3'}^2 + m_{3'}^2} \\
\Rightarrow \sqrt{p_1^2 + m_1^2} + \sqrt{p_2^2 + m_2^2} &= \sqrt{p_{1'}^2 + m_{1'}^2} + \sqrt{(\mathbf{p}_1 + \mathbf{p}_2 - \mathbf{p}_{1'} - \mathbf{p}_{3'})^2 + m_{2'}^2} + \sqrt{p_{3'}^2 + m_{3'}^2}.
\end{aligned} \tag{1.5}$$

We would like to solve eqn. (1.5) for  $p_{3'}$ , but there is a problem coming from the fact that  $p_{3'}$  appears under two square root symbols which makes it impossible to get an expression of the form  $p_{3'} = \dots$ . Instead, we will make the approximation that because  $p_{3'} \ll p_i$  for  $i \in \{1, 2, 1', 2'\}$  that we can ignore the  $p_{3'}$  that shows up in  $\sqrt{(\mathbf{p}_1 + \mathbf{p}_2 - \mathbf{p}_{1'} - \mathbf{p}_{3'})^2 + m_{2'}^2}$ . The result we obtain when solving for  $p_{3'}$  is then

$$\begin{aligned}
p_{3'} &= \left[ m_1^2 + m_{1'}^2 + m_2^2 + m_{2'}^2 - m_{3'}^2 + 2p_1^2 - 2p_1 p_{1'} c_{11'} + 2p_{1'}^2 - 2E_1 E_{1'} \right. \\
&\quad + 2p_1 p_2 c_{12} - 2p_2 p_{1'} c_{21'} + 2p_2^2 + 2E_1 - 2E_1 E_2 \\
&\quad - 2E_1 E_{2'}(p_1, p_2, p_{1'}, c_{11'}, c_{12}, c_{21'}) + 2E_{1'} E_{2'}(p_1, p_2, p_{1'}, c_{11'}, c_{12}, c_{21'}) \\
&\quad \left. - 2E_2 E_{2'}(p_1, p_2, p_{1'}, c_{11'}, c_{12}, c_{21'}) \right]^{1/2}
\end{aligned} \tag{1.6}$$

where  $E_i \equiv \sqrt{p_i^2 + m_i^2}$ ,  $c_{ij} \equiv \cos \theta_{ij} \equiv \mathbf{p}_i \cdot \mathbf{p}_j / (p_i p_j)$  and

$$E_{2'}(p_1, p_2, p_{1'}, c_{11'}, c_{12}, c_{21'}) = \sqrt{p_1^2 - 2p_1 p_{1'} c_{11'} + p_{1'}^2 + 2p_1 p_2 c_{12} - 2p_2 p_{1'} c_{21'} + p_2^2 + m_{2'}^2}. \tag{1.7}$$

Now that we have an expression for  $p_{3'}$  the requirement that  $p_{3'}$  must be real and non-negative restricts the domain of integration of the other variables. I am not sure how to derive the new limits so I got stuck here.

## 1.2 A recursive expression for phase space integrals

The problem I ran into at the end of the last section was how to use the energy-momentum delta functions to restrict the bounds of integration. I spent some time searching the literature for useful references and the most promising were refs. [1, 2, 3]. [1] provides an introduction to Monte Carlo methods and discusses the phase space measure in sec. 9, which covers several ways to deal with the Dirac delta in the phase space integration measure by doing appropriate changes of variables. The most useful method, is covered in sec. 9.6 and is the same one used in [2]. Ref. [2] extends the treatment in sec. 9.6 of [1] by also introducing momentum transfer variables. My understanding is that this is appropriate when your matrix element is flat in terms of the momentum transfers.

Consider the phase space integral  $R_n(p_a + p_b)$  defined by

**Phase space integral (without  $2\pi$  factors)**

$$R_n(p_a + p_b) = \int \cdots \int \prod_{i=1}^n \delta(p_i^2 - m_i^2) \Theta(p_i^0) d^4 p_i \delta^4(p_a + p_b - p_1 - \cdots - p_n). \quad (1.8)$$

where  $p_i$  now denotes a four-momentum vector in contrast to the previous section where it represented a three-momentum magnitude. This is manifestly Lorentz invariant and therefore  $R_n$  may only be a function of  $s \equiv (p_a + p_b)^2$ .

We note that  $R_n(s)$  is related to  $d\Phi_n(s)$  by

$$R_n(s) = (2\pi)^{3n-4} \int d\Phi_n(s). \quad (1.9)$$

Define the new integration variable  $M_{n-1}^2 \equiv (p_a + p_b - p_n)^2$ . The physical significance of  $M_{n-1}$  is that it is the invariant mass of the first  $n-1$  particles, which can be seen using four-momentum conservation:  $p_a + p_b - p_n = p_1 + p_2 + \cdots + p_{n-1}$ . It is possible to show that the following recursive relation holds (eqn. (3) of [2])

$$R_n(s) = \int_{(m_1+m_2+\cdots+m_{n-1})^2}^{(\sqrt{s}-m_n)^2} dM_{n-1}^2 \int d\Omega_n \frac{\sqrt{\lambda(s, M_{n-1}^2, m_n^2)}}{8s} R_{n-1}(M_{n-1}^2) \quad (1.10)$$

where  $\lambda(x, y, z) = x^2 + y^2 + z^2 - 2xy - 2xz - 2yz$  and the solid angle  $d\Omega_n \equiv d\cos\theta_n d\varphi_n$  defines the direction of  $\mathbf{p}_n$  in the frame where  $p_a + p_b = (\sqrt{s}, \mathbf{0})$ . The upper limit is obtained by requiring that

$$|\mathbf{p}_n|^2 = \frac{\lambda(s, M_{n-1}^2, m_n^2)}{4s} \quad (1.11)$$

be positive whereas the lower limit is threshold below which  $R_{n-1}(M_{n-1}^2) = 0$ . Repeated application of eq. (1.10) yields

$$\begin{aligned} R_n(s) &= \int_{(m_1+m_2+\cdots+m_{n-1})^2}^{(\sqrt{s}-m_n)^2} dM_{n-1}^2 \int d\Omega_n \frac{\sqrt{\lambda(s, M_{n-1}^2, m_n^2)}}{8s} \\ &\times \int_{(m_1+m_2+\cdots+m_{n-2})^2}^{(M_{n-1}-m_{n-1})^2} dM_{n-2}^2 \int d\Omega_{n-1} \frac{\sqrt{\lambda(M_{n-1}^2, M_{n-2}^2, m_{n-1}^2)}}{8M_{n-1}^2} \\ &\times \cdots \times \int_{(m_1+m_2)^2}^{(M_3-m_3)^2} dM_2^2 \int d\Omega_3 \frac{\sqrt{\lambda(M_3^2, M_2^2, m_3^2)}}{8M_3^2} \times \int d\Omega_2 \frac{\sqrt{\lambda(M_2^2, m_1^2, m_2^2)}}{8M_2^2}. \end{aligned} \quad (1.12)$$

The important case for me is when  $n = 3$ :

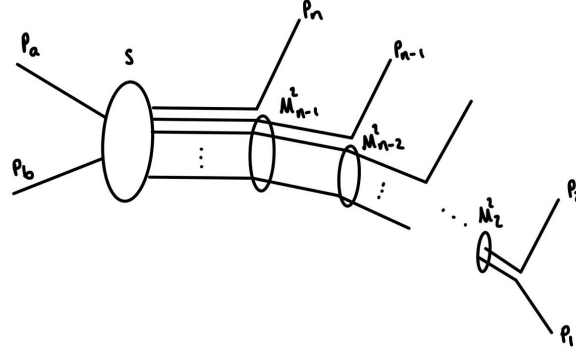


Figure 1: Illustration of the recursion relation as a sequence of effective  $2 \rightarrow 2$  scattering events (this diagram is largely based on pg. 26 of [1]).

#### Phase space measure for $n = 3$

$$\begin{aligned}
 R_3(s) &= (2\pi)^5 \int d\Phi_3(s) \\
 &= \int_{(m_1+m_2)^2}^{(\sqrt{s}-m_3)^2} dM_2^2 \int d\Omega_3 \frac{\sqrt{\lambda(s, M_2^2, m_3^2)}}{8s} \int d\Omega_2 \frac{\sqrt{\lambda(M_2^2, m_1^2, m_2^2)}}{8M_2^2}. \quad (1.13)
 \end{aligned}$$

It's easy to verify that the number of integration variables matches our expectation of  $3(3) - 4 = 5$  since we are integrating over one invariant mass, and two pairs of two angles.

In the above discussion we have considered  $p_a$  and  $p_b$  to be fixed. However, for my purposes I will also be integrating over  $p_a$  and  $p_b$ . This case seems to be similar to the one considered in [3] (cf. the example given between their eqs. (12) and (13)).

### 1.3 Writing the integral in terms of momentum transfers

In sec 1.2 we wrote the phase space integral in terms of invariant masses and angles of the three-momenta  $\mathbf{p}_i$  defined in the center-of-mass frame where  $\sum_{k=1}^i p_k = (M_i, \mathbf{0})$ . The matrix element in eq. (1.3) is a function of the Lorentz invariants  $(p_a \cdot p_1)$ ,  $(p_b \cdot p_3)$ ,  $(p_2 \cdot p_3)$ , and  $(p_b \cdot p_2) = -(p_a - p_1 - p_2 - p_3) \cdot p_2$ . It is cumbersome to write this matrix element explicitly in terms of the angles  $\{(\theta_i, \varphi_i) : i = 1, 2, \dots, n\}$  so we would rather use kinematic Lorentz invariants analogous to the Mandelstam variables for  $2 \rightarrow 2$  scattering<sup>3</sup>. To this end it is convenient to introduce the so-called ‘momentum transfers’  $t_i \equiv Q_i^2 \equiv (p_a - p_1 - \dots - p_i)^2 = (p_n + p_{n-1} + \dots + p_{i+1} - p_b)^2$ .

Some of the dot products (but not all) may be written exclusively in terms of the kinematic

<sup>2</sup>In eq. (3) of [2] the authors allude that the angular  $d\Omega$  integrals can be performed. They do not perform the integrals because “the variables defining a momentum vector should appear explicitly if a Monte Carlo method is to be applied for the generation of events.” In my case I think that we still can’t do these integrals trivially because the matrix element depends on them, although I’m not 100% sure if this is correct.

<sup>3</sup>At least, I think that is the motivation for introducing these variables.

variables. For example,

$$\begin{aligned}
t_1 &\equiv (p_a - p_1)^2 && \rightarrow && 2p_a \cdot p_1 = m_a^2 + m_1^2 - t_1 \\
t_2 &\equiv (p_a - (p_1 + p_2))^2 && \rightarrow && 2p_a \cdot p_2 = M_2^2 - m_1^2 + t_1 - t_2 \\
t_3 &\equiv (p_a - (p_1 + p_2 + p_3))^2 && \rightarrow && 2p_a \cdot p_3 = M_3^2 - M_2^2 + t_2 \\
M_2^2 &\equiv (p_1 + p_2)^2 && \rightarrow && 2p_1 \cdot p_2 = M_2^2 - m_1^2 - m_2^2 \\
M_3^2 &\equiv (p_1 + p_2 + p_3)^2 && \rightarrow && 2(p_1 + p_2) \cdot p_3 = M_3^2 - 2M_2^2 + m_1^2 + m_2^2 - m_3^2 \\
2p_b \cdot p_3 &= 2(p_1 + p_2) \cdot p_3 + 2m_3^2 - 2p_a \cdot p_3 = m_1^2 + m_2^2 + m_3^2 - M_2^2 - t_2
\end{aligned}$$

However I was unable to write  $p_2 \cdot p_3$  in terms of the invariants alone.  $p_b \cdot p_2$  is really just  $p_2 \cdot p_3$  in disguise so it has the same problem. I believe this is because these variables are not enough to fully describe the system and some angles are required in addition. We need  $3(3) - 4 = 5$  integration variables but only 3 of them are prescribed by the kinematic invariants  $M_2^2$ ,  $t_1$ , and  $t_2$ <sup>4</sup>. I haven't followed their derivation through in enough detail to understand precisely how they show up in the expression. The two additional degrees of freedom can be seen in eq. (14) of [2]. They are the azimuthal angles  $\varphi_1$ ,  $\varphi_2$  as seen in a frame where  $p_a = (m_a, \mathbf{0})$ .

To transform  $R_n$  into a form in which the momentum transfers appear as variables we must consider the phase space integral as a function of  $p_a$  and  $-p_b \equiv Q_n$  separately:  $R_n = R_n(p_a, Q_n)$ . Let  $Q_i = p_a - p_1 - \dots - p_i$  so that  $p_i = Q_{i-1} - Q_i$ . Then, eq. (1.8) becomes

$$\begin{aligned}
R_n(s) &= \int \left[ \prod_{i=1}^n d^4 Q_i \delta((Q_{i-1} - Q_i)^2 - m_i^2) \right] \delta^4(p_a + p_b - \sum_{i=1}^n (Q_{i-1} - Q_i))|_{Q_0 \equiv p_a} \\
&= \int \left[ \prod_{i=1}^n d^4 Q_i \delta((Q_{i-1} - Q_i)^2 - m_i^2) \right] \delta^4(Q_n + p_b) \\
&= \int d^4 Q_{n-1} \delta((Q_{n-2} - Q_{n-1})^2 - m_{n-1}^2) \left[ \prod_{i=1}^{n-2} d^4 Q_i \delta((Q_{i-1} - Q_i)^2 - m_i^2) \right].
\end{aligned} \tag{1.14}$$

Understand how to write the matrix element explicitly in terms of the angles  $\varphi_1$ ,  $\varphi_2$  as well as the invariants  $M_2^2$ ,  $t_1$ , and  $t_2$ .

In eq. (1.14) I didn't write the Heaviside function which picks out positive energies, just pretend it is there so that  $(Q_{i-1} - Q_i)^0 > 0$ .

The recursion relation is then simply

$$R_n(p_a, Q_n) = \int d^4 Q_{n-1} \delta((Q_{n-1} - Q_n)^2 - m_n^2) R_{n-1}(p_a, Q_{n-1}). \tag{1.15}$$

$R_n$  is now regarded as a function of two invariants  $s = s_n = (p_a - Q_n)^2$ , and  $t_n = Q_n^2$ :  $R_n = R_n(p_a, Q_n) = R_n(s_n, t_n)$ . Introducing these variables in eq. (1.15) the authors of [2] obtain

$$R_n(s_n, t_n, s_{n-1}, t_{n-1}) = \int ds_{n-1} dt_{n-1} K(s_n, t_n, s_{n-1}, t_{n-1}) R_{n-1}(s_{n-1}, t_{n-1}), \tag{1.16}$$

where

$$K(s_n, t_n, s_{n-1}, t_{n-1}) \tag{1.17}$$

$$= \int d^4 Q_{n-1} \delta(Q_{n-1}^2 - t_{n-1}) \delta(s_{n-1} - (p_a - Q_{n-1})^2) \delta((Q_{n-1} - Q_n)^2 - m_n^2) \tag{1.18}$$

$$= \int_0^{2\pi} d\varphi_{n-1} \frac{1}{4\sqrt{\lambda(s_n, t_n, m_a^2)}} \Theta(-G(t_{n-1}, s_n, s_{n-1}, t_n, m_n^2, m_a^2)), \tag{1.19}$$

$\Theta(x)$  is the Heaviside step-function, and  $\varphi_{n-1}$  defines the azimuthal angle of  $\mathbf{Q}_{n-1}$  in the frame  $p_a = (m_a, \mathbf{0})$ .  $G$  is a kinematic function defined by

$$G(x, y, z, u, v, w) = -\frac{1}{2} \left| \begin{array}{ccc} 2u & x+u-v & u+w-y \\ & 2x & x-z+w \\ \text{(symm.)} & & 2w \end{array} \right| \quad (1.20)$$

$$= x^2y + xy^2 + z^2u + vw^2 + v^2w$$

$$+ xzw + xuv + yzv + yuw$$

$$- xy(z+u+v+w) - zu(x+y+v+w) - vw(x+y+z+u) .$$

$R_n$  for  $n = 3$

$$R_3(s_3, t_3) = \frac{1}{4\sqrt{\lambda(s_3, t_3, m_a^2)}} \int \frac{ds_2 dt_2 d\varphi_2}{4\sqrt{\lambda(s_2, t_2, m_a^2)}} \Theta(-G_2) \int dt_1 d\varphi_1 \Theta(-G_1) \quad (1.21)$$

with

$$G_i = G(t_i, s_{i+1}, s_i, t_{i+1}, m_{i+1}^2, m_a^2) . \quad (1.22)$$

## 2 Monte Carlo integration

### 2.1 The basics

Suppose you are interested in evaluating the integral

$$I = \int_a^b f(x) dx. \quad (2.1)$$

The most obvious method of approximating this is with a Riemann sum or trapezoidal integration where you divide the domain into bins of width  $\Delta x$  and calculate the area of a rectangle or trapezoid using the values of  $f$  at the bin edges.

Monte Carlo integration, on the other hand, takes a different approach altogether whereby points in the domain are sampled at random and the integral is approximated as an average. To see this, note that the average value of  $f$  on the domain is

$$\langle f \rangle = \frac{1}{b-a} \int_a^b f(x) dx = \frac{I}{b-a}. \quad (2.2)$$

The average  $\langle f \rangle$  can be approximated by uniformly sampling  $N$  i.i.d random numbers  $x_i$  on the domain  $x \in [a, b]$  where  $i \in 1, 2, \dots, N$  and calculating the mean of  $f(x_i)$ :

$$\langle f \rangle \approx \langle f \rangle_N \equiv \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (2.3)$$

---

<sup>4</sup> $M_1^2 = m_1^2$ ,  $M_3^2 = s$  and  $t_3 = (-p_b)^2 = m_b^2$  are all constants, not integration variables.

Therefore we can approximate the integral  $I$  as

$$I \approx (b - a) \langle f \rangle_N. \quad (2.4)$$

This argument can be extended to higher dimensions, i.e. in  $n$  dimensions we have

$$I = \int_V f(\mathbf{x}) d^n x = V \langle f \rangle. \quad (2.5)$$

One important result is that this strategy always converges as  $N^{-1/2}$  in all dimensions, whereas techniques like trapezoidal integration converge rapidly in one dimension ( $N^{-2}$ ), but significantly slower for higher dimensions ( $N^{-2/n}$ ). For example in 5 dimensions, trapezoidal integration converges like  $N^{-2/5}$ , which is slower than Monte Carlo integration.

## 2.2 Importance sampling

In sec. 2.1 we sampled  $x$  uniformly at random on the domain of integration. However, we can choose to sample  $x$  from any arbitrary distribution  $p(x)$  and compute the same expectation value by weighting the sample  $f(x)$  by  $1/p(x)$  so that

$$I \approx \frac{V}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}. \quad (2.6)$$

With an appropriate choice of  $p$  our Monte Carlo integration technique can be made to converge significantly faster than by sampling  $x$  uniformly at random. To see why this is, suppose  $f(x) = e^{-x^2}$  and  $[a, b] = [-1000, 1000]$ . Sampling uniformly on  $[-1000, 1000]$  means that the majority of points we pick will be in the tails of the Gaussian and will hardly affect the value of the integral. If instead we sampled from a standard normal distribution we would mostly choose  $x$  values near the peak at  $x = 0$  and our result would converge much more rapidly. Clearly the choice of  $p$  is heavily dependent on the particular shape of the integrand  $f$ . The Metropolis-Hastings algorithm is one way to generate samples  $x$  from  $p(x)$  and can be used for importance sampling.

VEGAS algorithm:

The discussion of the VEGAS algorithm at [https://www.ippp.dur.ac.uk/~krauss/Lectures/QuarksLeptons/Basics/PS\\_Vegas.html](https://www.ippp.dur.ac.uk/~krauss/Lectures/QuarksLeptons/Basics/PS_Vegas.html) is quite good. In case the link breaks in the future I reproduce some of the main points below.

“The VEGAS algorithm starts by dividing the  $n$ -dimensional hypercube  $[0, 1]^n$  into smaller ones – usually of identical size – and performs a sampling in each of them. The results are then used to refine the integration grid for the next iteration of the sampling procedure.”



## 2.3 Stratified sampling

## 2.4 Multichannel sampling

# 3 CompHEP and MADGRAPH5

## 3.1 CompHEP

While I was looking for ways to do the integral I found a software called CompHEP [4] which does a lot of the things I am interested in doing. CompHEP is a “a package for evaluation of Feynman diagrams, integration over multi-particle phase space and event generation”. It allows the user to specify a Lagrangian and then it can generate tree level Feynman diagrams for a user-specified process, along with C and Fortran code to evaluate matrix elements. It can even compute cross-sections/other observables. CompHEP has two main backbones: (i) symbolic calculations (ii) numerical calculations.

The main tasks performed by the symbolic part are

1. Select a process by specifying in and out states for decays into up to 5 outgoing particles or collisions of  $2 \rightarrow 2$ ,  $2 \rightarrow 3$ ,  $2 \rightarrow 4$ .
2. Generate Feynman diagrams.
3. Delete unwanted diagrams.
4. Compute S-matrix elements.
5. Save symbolic results so they can be further analysed in *Mathematica*.
6. Generate C and Fortran code for the matrix elements to be used in numerical studies.

The numerical part, on the other hand, performs Monte Carlo integration and event generation. The main tasks performed are:

1. Choose phase space kinematic variables (idk what this means yet)
2. Introduce kinematic cuts over squared momentum transfers (not sure what the significance of this is)
3. Perform regularization to remove sharp peaks in matrix elements.
4. Calculate distributions, cross-sections, or particle width using Monte Carlo methods.
5. Perform integration, taking into account structure functions for ingoing particles.
6. Event generation.

## A Derivation of recursive relation

This appendix fills in the details for the derivation of eq. (1.10) which are left out in [2]. Our starting point is

$$\begin{aligned}
 R_n(s) &= \int \prod_{i=1}^n \delta(p_i^2 - m_i^2) \Theta(p_i^0) d^4 p_i \delta^4(p_a + p_b - p_1 - \cdots - p_n) \\
 &= \int \delta(p_n^2 - m_n^2) \Theta(p_n^0) d^4 p_n \left[ \prod_{i=1}^{n-1} \delta(p_i^2 - m_i^2) \Theta(p_i^0) \right] \delta^4(p_a + p_b - p_n - p_1 - \cdots - p_{n-1}) \\
 &= \int \delta(p_n^2 - m_n^2) \Theta(p_n^0) d^4 p_n R_{n-1}(p_a + p_b - p_n)
 \end{aligned}$$

Multiply the RHS by  $\int dM_{n-1}^2 \delta(M_{n-1}^2 - (p_a + p_b - p_n)^2)$ .

$$\int dM_{n-1}^2 \delta(M_{n-1}^2 - (p_a + p_b - p_n)^2) \delta(p_n^2 - m_n^2) \Theta(p_n^0) d^4 p_n R_{n-1}(p_a + p_b - p_n).$$

Use the on-shell delta function to do the  $p_n^0$  integral and write  $d^3 p_n$  in terms of spherical coordinates:

$$\begin{aligned}
 &\int dM_{n-1}^2 \delta(M_{n-1}^2 - (p_a + p_b - p_n)^2) \frac{d^3 p_n}{2E_n} R_{n-1}(p_a + p_b - p_n) \\
 &= \int dM_{n-1}^2 \delta(M_{n-1}^2 - (p_a + p_b - p_n)^2) \frac{|\mathbf{p}_n|^2 d|\mathbf{p}_n| d\Omega_n}{2E_n} R_{n-1}(p_a + p_b - p_n).
 \end{aligned}$$

Then use the  $M_{n-1}^2$  Dirac delta to do the  $d|\mathbf{p}_n|$  integral. This requires us to (i) solve  $M_{n-1}^2 - (p_a + p_b - p_n)^2 = 0$  for  $|\mathbf{p}_n|$ ; (ii) evaluate  $d/d|\mathbf{p}_n|(M_{n-1}^2 - (p_a + p_b - p_n)^2)$ ; and (iii) impose  $|\mathbf{p}_n|^2 \geq 0$  to derive bounds of integration for  $\int dM_{n-1}^2$ .

(i) We start by expanding the LHS

$$\begin{aligned}
 M_{n-1}^2 - (p_a + p_b - p_n)^2 &= M_{n-1}^2 - (p_a + p_b)^2 - p_n^2 + 2(p_a + p_b) \cdot p_n \\
 &= M_{n-1}^2 - s - m_n^2 + 2(E_a + E_b)E_n + 2(\mathbf{p}_a + \mathbf{p}_b) \cdot \mathbf{p}_n.
 \end{aligned}$$

Now we choose the center of mass frame so that  $\mathbf{p}_a + \mathbf{p}_b = 0$ . This is valid because the integral is a Lorentz invariant, so the final result is independent of the reference frame.

$$\begin{aligned}
 &M_{n-1}^2 - s - m_n^2 + 2(E_a + E_b)E_n + 2(\mathbf{p}_a + \mathbf{p}_b) \cdot \mathbf{p}_n \\
 &= M_{n-1}^2 - s - m_n^2 + 2\sqrt{s}\sqrt{|\mathbf{p}_n|^2 + m_n^2} = 0.
 \end{aligned}$$

Now we can rearrange and solve for  $|\mathbf{p}_n|$

$$\begin{aligned}
 |\mathbf{p}_n|^2 &= \frac{(s + m_n^2 - M_{n-1}^2)^2}{4s} - m_n^2 = \frac{s^2 + m_n^4 + M_{n-1}^4 - 2sm_n^2 - 2sM_{n-1}^2 - 2m_n^2 M_{n-1}^2}{4s} \\
 &\equiv \frac{\lambda(s, M_{n-1}^2, m_n^2)}{4s}.
 \end{aligned}$$

$\lambda(x, y, z) = x^2 + y^2 + z^2 - 2xy - 2xz - 2yz$  is called the Källén function.

(ii)

$$\begin{aligned}
& \frac{d}{d|\mathbf{p}_n|} \left( M_{n-1}^2 - (p_a + p_b - p_n)^2 \right) \\
&= \frac{d}{d|\mathbf{p}_n|} \left( M_{n-1}^2 - s - m_n^2 + 2\sqrt{s} \sqrt{|\mathbf{p}_n|^2 + m_n^2} \right) \\
&= \boxed{\frac{2\sqrt{s} |\mathbf{p}_n|}{E_n}}
\end{aligned}$$

(iii)

$$\begin{aligned}
& |\mathbf{p}_n|^2 \geq 0 \\
& \Rightarrow \lambda(s, M_{n-1}^2, m_n^2) \geq 0 \\
& \Rightarrow [M_{n-1}^2 - (s + m_n^2)]^2 - 4sm_n^2 \geq 0 \\
& \Rightarrow [M_{n-1}^2 - (s + m_n^2)]^2 \geq 4sm_n^2 \\
& \Rightarrow M_{n-1}^2 - (s + m_n^2) \geq 2\sqrt{s m_n^2} \parallel M_{n-1}^2 - (s + m_n^2) \leq -2\sqrt{s m_n^2} \\
& \Rightarrow M_{n-1}^2 \geq (s + m_n^2) + 2\sqrt{s m_n^2} \parallel M_{n-1}^2 \leq (s + m_n^2) - 2\sqrt{s m_n^2} \\
& \Rightarrow \boxed{M_{n-1}^2 \geq (\sqrt{s} + m_n)^2 \parallel M_{n-1}^2 \leq (\sqrt{s} - m_n)^2}.
\end{aligned}$$

With these results established we find that

$$\delta(M_{n-1}^2 - (p_a + p_b - p_n)^2) = \frac{\delta(|\mathbf{p}_n| - \sqrt{\lambda(s, M_{n-1}^2, m_n^2)}) / (2\sqrt{s})}{2\sqrt{s} |\mathbf{p}_n| / E_n}. \quad (1.1)$$

Hence,

$$\begin{aligned}
& \int dM_{n-1}^2 \delta(M_{n-1}^2 - (p_a + p_b - p_n)^2) \frac{|\mathbf{p}_n|^2 d|\mathbf{p}_n| d\Omega_n}{2E_n} R_{n-1}(p_a + p_b - p_n) \\
&= \int dM_{n-1}^2 \frac{\delta(|\mathbf{p}_n| - \sqrt{\lambda(s, M_{n-1}^2, m_n^2)}) / (2\sqrt{s})}{2\sqrt{s} |\mathbf{p}_n| / E_n} \frac{|\mathbf{p}_n|^2 d|\mathbf{p}_n| d\Omega_n}{2E_n} R_{n-1}(p_a + p_b - p_n) \\
&= \int dM_{n-1}^2 \frac{\sqrt{\lambda(s, M_{n-1}^2, m_n^2)}}{8s} d\Omega_n R_{n-1}(M_{n-1}^2).
\end{aligned}$$

## References

- [1] F. James. Monte-Carlo phase space. 5 1968.
- [2] E. Byckling and K. Kajantie. N-particle phase space in terms of invariant momentum transfers. *Nucl. Phys. B*, 9:568–576, 1969.

- 
- [3] Joshua Isaacson, Stefan H  che, Diego Lopez Gutierrez, and Noemi Rocco. Novel event generator for the automated simulation of neutrino scattering. *Phys. Rev. D*, 105(9):096006, 2022.
  - [4] E. E. Boos, M. N. Dubinin, V. A. Ilyin, A. E. Pukhov, and V. I. Savrin. CompHEP: Specialized package for automatic calculations of elementary particle decays and collisions. 10 1994.