qtAMEG: Automatic Matrix Element Generation for Tree-level QED

Tony Menzo

Abstract

The quantum electrodynamics (QED) tree-level automatic matrix element generator (qtAMEG) generates all contributing Feynman diagrams for $2 \to 2$ tree-level processes. The program allows the user to input any allowed $2 \to 2$ QED (polarized or unpolarized) process along with a specified solid angle and receive an estimation for the differential scattering cross section.

1 Introduction

The calculation of the total scattering cross section is a marquee computation in high energy particle physics. The cross section bridges the gap between pen and paper or, in this case, source code and computer and collider experiments. It is the ultimate probe and testing ground to our theoretical understanding of fundamental high energy physics.

The theoretical calculation of the total cross section by hand is a long and tedious process prone to errors. Particularly, the most laborious process is the generation and evaluation of all contributing matrix elements or Feynman diagrams. Tree-level calculations, if one is proficient, can be done optimistically in under an hour. Second order and higher calculations however, become extremely laborious and can take days, weeks, or even months to compute the contributions of just one diagram. Not only do the individual diagram calculations become extremely computationally complex but the number of diagrams with each successive order increases significantly [5][6]. For example, computing contributions to the electrons gyromagnetic ratio to tenth order in perturbation theory involves the evaluation of 12,672 diagrams [7]. These problems segue into the need for algorithmic generation and evaluation of Feynman diagrams and matrix elements.

There are a number of already existing software packages such as `MadGraph` [9] and `WHIZARD` [8] which tackle the problem of automatically generating and evaluating matrix elements at tree level and next leading order (NLO). My quantum electrodynamic tree-level automatic matrix element generator or `qtAMEG` aims to generate contributing matrix elements for $2 \to 2$ tree-level QED processes in order to obtain differential scattering cross sections.

2 Theory [1][2]

2.1 QED

Quantum electrodynamics (QED), a relativistic quantum field theory that describes the electromagnetic interaction of quarks and leptons, is one of the most accurate physical theories in Physics. With determinations of the fine structure constant α via field-theoretic calculation in QED and experiment agreeing to roughly one part per million[4].

$$\alpha^{-1}_{\text{exp}} = 137.03599878(91), \qquad \alpha^{-1}_{\text{theory}} = 137.035999710(96) \tag{2.1}$$

$$\alpha_{\text{exp}}^{-1} - \alpha_{\text{theory}}^{-1} = -0.93(0.92) \times 10^{-6} \tag{2.2}$$

In `qtAMEG` the calculation of the transition amplitudes are based upon a perturbation expansion in powers of the fine structure constant α. A detailed explanation of these calculations is out of the scope of this manual, to remain self-contained I will outline the calculation below.

2.2 $\mathcal{S}$-matrix

The evolution of a free particle state with energy E from $t = t_1$ to $t = t_2$ is given by the unitary time evolution operator

$$\mathcal{U}(t_2, t_1) = \exp\left[-i(t_2 - t_1)E\right] \tag{2.3}$$

The time evolution of a scattering process can be effectively described by an object called the scattering matrix ($\mathcal{S}$-matrix). If we assume that our interactions occur at short distances then the whole of the interaction takes place in a finite time frame $-t_0/2 < t < t_0/2$. The $\mathcal{S}$-matrix is described as the product of unitary time evolution operators

$$\mathcal{S}_{fi} = \mathcal{U}_f^{-1}(t_0/2, 0)\mathcal{U}_{fi}(t_0/2, -t_0/2)\mathcal{U}_i^{-1}(0, -t_0/2) \tag{2.4}$$

Where the initial and final free particle unitary time evolution operators are given by,

$$\mathcal{U}_i(0, -t_0) = \exp\left[-it_0/2 \sum_{i=1}^{n_i} E_i\right], \qquad \mathcal{U}_f(t_0, 0) = \exp\left[-it_0/2 \sum_{j=1}^{n_f} E_j\right] \tag{2.5}$$

The sums runs over the n_i initial particles and n_f final particle states. Given an initial state $|i\rangle$ and final state $|f\rangle$ we can write the $\mathcal{S}_{fi}$ matrix element in terms of a transition element $\mathcal{T}_{fi}$ and an identity element corresponding to no scattering

$$\mathcal{S}_{fi} = \langle f|i\rangle + i(2\pi)^4\delta^4(p_i - p_f)\mathcal{T}_{fi} \tag{2.6}$$

Where the four-dimensional delta function ensures energy and momentum conservation. The transition probability P_{fi} is given by the modulus of the $\mathcal{S}$-matrix element, for $i \neq f$ we have

$$P_{fi} = |\mathcal{S}_{fi}|^2 = [(2\pi)^4\delta^4(p_i - p_f)]^2|\mathcal{T}_{fi}|^2 = t_0V(2\pi)^4\delta^4(p_i - p_f)|\mathcal{T}_{fi}|^2 \tag{2.7}$$

Where V is the volume of our imaginary enclosed box.

The total scattering cross section for two body scattering $a+b \to n$ scattering process is given by

$$\sigma_{2\to n} = \frac{N_n}{2t_0 V\sqrt{\lambda(s, m_a^2, m_b^2)}} \tag{2.8}$$

Where N_n is the total number of events detected in the detector, and

$$\lambda(s, m_a^2, m_b^2) = \left(s - (m_a + m_b)^2\right)\left(s - (m_a - m_b)^2\right) \tag{2.9}$$

$$s = (E_a + E_b)^2 = (p_a + p_b)^2 \tag{2.10}$$

The total number of events in the detector N_n is defined as

$$N_n = \int |S_{ab\to n}|^2 d\text{Lips}(p_1, p_2, \cdots, p_n) \tag{2.11}$$

Where the Lorentz invariant phase space is defined as

$$d\text{Lips} \equiv \frac{1}{(2\pi)^{3n}} \prod_{i=1}^{n} \frac{d^3\mathbf{p}}{2E_i} \tag{2.12}$$

Thus, the scattering cross section is expressed in terms of transition amplitudes is given by

$$\sigma_{ab\to n} = \frac{1}{2\sqrt{\lambda(s, m_a^2, m_b^2)}} \int (2\pi)^4 \delta^4(p_i - p_f) d\text{Lips}(p_1, p_2, \cdots, p_n) |T_{fi}|^2 \tag{2.13}$$

Where $p_i = p_a + p_b$ and $p_f = \sum_{j=1}^{n} p_{f,j}$

We see explicitly now that the calculation of the total cross section can be split into two pieces:

1. Determine the size of our kinematically available phase space
2. Determine the transition amplitude matrix elements T_{fi} via the Feynman Rules

2.3 Kinematics

First we need to define the kinematics of the problem. We imagine two particles speeding towards each other and colliding. We define the following outgoing 4-momentum

$$p_1' = (E_1', \mathbf{p_1'}), \qquad p_2' = (E_2', \mathbf{p_2'}), \qquad p_3 = (E_3, \mathbf{p}_3), \qquad p_4 = (E_4, \mathbf{p}_4) \tag{2.14}$$

$$P' \equiv (p_1' + p_2'), \qquad P'^2 = (p_1' + p_2')^2 = (p_3 + p_4)^2 \equiv s \tag{2.15}$$

In the center-of-mass frame we have

$$\mathbf{p'}_1 = \mathbf{p} = -\mathbf{p'}_2 \tag{2.16}$$

$$s = (E_1' + E_2')^2 \tag{2.17}$$

Thus,

$$s = (p_1' + p_2')^2 = m_1^2 + m_2^2 + 2p_1' \cdot p_2' \tag{2.18}$$

$$p_1' \cdot p_2' = E_1' E_2' - \mathbf{p_1'} \cdot \mathbf{p_2'} = E_1' E_2' + |\mathbf{p'}|^2 \tag{2.19}$$

Plugging back into Eq.(2.18)

$$s = m_1^2 + m_2^2 + 2(E_1 E_2 + |\mathbf{p'}|^2) \tag{2.20}$$

$$(s - m_1^2 - m_2^2) = 2(E_1 E_2 + |\mathbf{p'}|^2) \tag{2.21}$$

Remembering that $E_1 = \sqrt{m_1^2 + |\mathbf{p'}|^2}$ and $E_2 = \sqrt{m_2^2 + |\mathbf{p'}|^2}$ we find that the following relation is true

$$|\mathbf{p'}|\sqrt{s} = \frac{1}{2}\sqrt{\lambda} \tag{2.22}$$

Now that we have all of the necessary kinematic information let's begin the calculation. We start with the definition of the infinitesimal Lorentz invariant phase space (dLips) of a system with center of mass energy s, and momentum $P_1, P_2, ..., P_n$

$$d\text{Lips}(s; p_1, \cdots, p_n) = (2\pi)^4 \delta^4(p_i - p_f) d\text{Lips}(p_1, \cdots, p_n) \tag{2.23}$$

For two body phase space we have

$$d\text{Lips}(s; p_1', p_2') = (2\pi)^4 \delta^4(p_1' + p_2' - p_3 - p_4) \frac{1}{(2\pi)^6} \frac{d^3\mathbf{p}_3}{2E_3} \frac{d^3\mathbf{p}_4}{2E_4} \tag{2.24}$$

Now, we can split up the four dimensional delta function into two delta functions. One which ensures conservation of momentum and the other which ensures conservation of energy.

$$= \frac{1}{16\pi^2} \delta^3(\mathbf{p} - \mathbf{p}_3 - \mathbf{p}_4) \delta(\sqrt{s} - E_3 + E_4) \frac{d^3\mathbf{p}_3}{E_3} \frac{d^3\mathbf{p}_4}{E_4} \tag{2.25}$$

In the center of mass frame $\mathbf{p} = 0$ so

$$= \frac{1}{16\pi^2} \delta^3(\mathbf{p_3} + \mathbf{p_4}) \delta(\sqrt{s} - E_3 + E_4) \frac{d^3\mathbf{p}_3}{E_3} \frac{d^3\mathbf{p}_4}{E_4} \tag{2.26}$$

In the center of mass frame we can now integrate over $\mathbf{p}_4$

$$= \frac{1}{16\pi^2}\delta(\sqrt{s} - E_3 + E_4)\frac{d^3\mathbf{p}}{E_3 E_4} \tag{2.27}$$

We can rewrite the momentum space volume differential

$$d^3\mathbf{p} = |\mathbf{p}|^2 \sin\theta d|\mathbf{p}|d\theta d\phi = p^2 dp d\Omega \tag{2.28}$$

Where $\sin\theta d\theta d\phi = d\Omega$ is the differential solid angle and $p \equiv |\mathbf{p}|$. Plugging this result into Eq.(2.27)

$$= \frac{1}{16\pi^2}\delta(\sqrt{s} - E_3 - E_4)\frac{p^2 dp d\Omega}{E_3 E_4} \tag{2.29}$$

We can rewrite the differential momentum in the following way

$$E = E_3 + E_4 = \sqrt{m_3^2 + p^2} + \sqrt{m_4^2 + p^2} \tag{2.30}$$

$$\frac{dE}{dp} = \frac{p}{\sqrt{m_3^2 + p^2}} + \frac{p}{\sqrt{m_4^2 + p^2}} = \frac{p}{E_3} + \frac{p}{E_4} = \frac{p(E_3 + E_4)}{E_3 E_4} \tag{2.31}$$

$$dp = \frac{E_3 E_4 dE}{pE} \tag{2.32}$$

Plugging back into Eq.(2.29) we get

$$= \frac{1}{16\pi^2}\delta(\sqrt{s} - E)\frac{p dE d\Omega}{E} \tag{2.33}$$

After integrating over E we are left with

$$d\text{Lips}(s; p_1', p_2') = \frac{1}{16\pi^2}\frac{p d\Omega}{\sqrt{s}} \tag{2.34}$$

Using Eq.(2.22) and defining $|\mathbf{p}'| \equiv p'$

$$d\text{Lips}(s; p_1, \cdots, p_n) = \frac{1}{8\pi^2}\frac{p}{p'}\frac{d\Omega}{\sqrt{\lambda}} \tag{2.35}$$

From Eq.(2.13), the differential cross section is now given by

$$\frac{d\sigma_{ab\to bc}}{d\Omega} = \frac{1}{2\sqrt{\lambda}}\frac{1}{8\pi^2}\frac{p}{p'}\frac{d\Omega}{\sqrt{\lambda}}|\mathcal{T}_{fi}|^2 = \frac{1}{64\pi^2 s}\frac{p}{p'}|\mathcal{T}_{fi}|^2 \tag{2.36}$$

Reinstating all dimensionful quantities we are left with

$$\frac{d\sigma_{ab\to bc}}{d\Omega} = \left(\frac{\hbar c}{8\pi}\right)^2 \frac{p}{p'}\frac{1}{s}|\mathcal{T}_{fi}|^2 \tag{2.37}$$

The in `qtAMEG` where both particles will be traveling with equal and opposite momenta oriented along the z-axis, the outgoing momenta can be parameterized as follows

$$p_3 = (E_1, q\sin\theta\cos\phi, q\sin\theta\sin\theta, q\cos\phi), \quad p_4 = (E_1, -q\sin\theta\cos\phi, -q\sin\theta\sin\theta, -q\cos\phi) \tag{2.38}$$

with, $q = \sqrt{E_1^2 - m_3^2}$.

All that's left is to determine the transition matrix elements.

2.4 Feynman Rules

In QED there is only one interaction vertex denoted $f\bar{f}\gamma$ (see Figure 1).

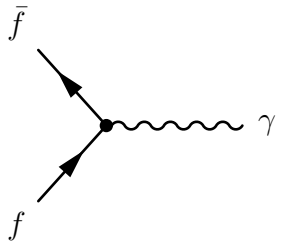
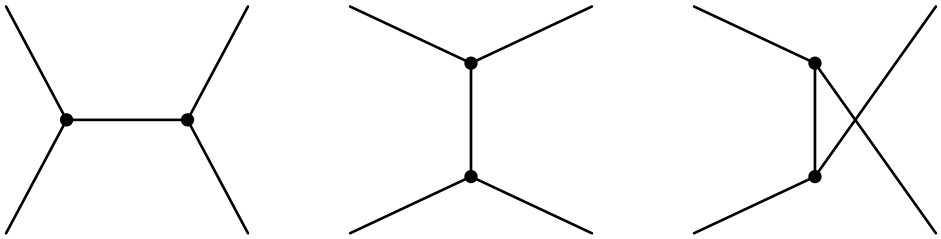


Figure 1: QED vertex

For $2 \to 2$ tree-level processes there are only three potential contributing diagrams in QED, namely s, t, and u-channel diagrams (Figure 2) corresponding to different transfers of momentum between incoming and outgoing particles.



s-channel t-channel u-channel

Figure 2: Feynman diagrams for s, t, and u-channel diagrams

To calculate all contributing transition amplitudes $\mathcal{T}_{fi}$ from Feynman diagrams for $2 \to 2$ scattering, we follow these steps

1. Draw all two-vertex diagrams allowed by the QED vertex (see Figure 1)

   There are six possible external lines in QED

   (a) Incoming particle ( ): u

   (b) Outgoing particle ( ): $\bar{u}$

   (c) Incoming antiparticle ( ): $\bar{v}$

   (d) Outgoing antiparticle ( ): v

   (e) Incoming photon ( ): ϵ_μ

   (f) Outgoing photon ( ): ϵ^*_μ

2. Associate each external and internal line with a momentum.

3. Associate each vertex with a factor of

$$ig\gamma^\mu \tag{2.39}$$

   Where γ^μ is the four-component vector of Dirac matrices and $g = -2q'\sqrt{\pi}$ where q' is the charge of the **particle** in the vertex.

4. The internal line contributions for photons and fermions are as follows

   For photons:

$$\frac{-ig_{\mu\nu}}{q^2} \tag{2.40}$$

   For fermions:

$$\frac{i(\gamma^\mu q_\mu + mc)}{q^2 - m^2c^2} \tag{2.41}$$

   Where q represents the momentum transferred from the external lines to internal line through the vertex. From energy and momentum conservation each of the respective diagrams in Figure 2 have corresponding q's according to

   s-channel:

$$q = s \equiv (p_1 + p_2)^2 = (p_3 + p_4)^2 \tag{2.42}$$

   t-channel:

$$q = t \equiv (p_1 - p_3)^2 = (p_2 - p_4)^2 \tag{2.43}$$

u-channel:

$$q = u \equiv (p_1 - p_4)^2 = (p_2 - p_3)^2 \tag{2.44}$$

5. Starting from an outgoing fermion line or incoming anti-fermion line, write down each external line, vertex, external line contribution as you trace backwards alone the fermion. Do this for all external lines and then include the internal line contribution, be careful and ensure indices are correctly matched.

6. If two diagrams differ by only the interchange of two incoming or outgoing fermions, add a relative minus sign between the two diagrams. This takes into account the antisymmetric property of identical fermions.

For example in Møller scattering ($e^-e^- \to e^-e^-$) there are two contributing diagrams, namely t and u-channels. The Feynman diagram for the t-channel is given in Figure 3

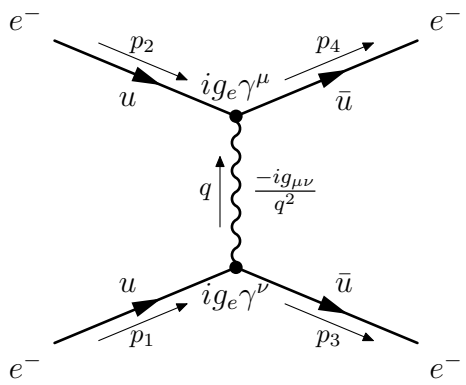


Figure 3: Example t-channel diagram for $e^-e^- \to e^-e^-$ scattering

The corresponding matrix element for Figure 3 is given by

$$\mathcal{T}_{e^-e^-,t} = \frac{-g_e^2}{(p_1 - p_3)^2}[\bar{u}(p_4)\gamma^\mu u(p_2)][\bar{u}(p_3)\gamma_\mu u(p_1)] \tag{2.45}$$

2.5 Gamma matrices, Spinors, Polarization vectors [3]

We follow the same conventions as in [3].

2.5.1 γ matrices

In the Weyl-basis the γ matrices are defined as follows

$$\gamma^\mu = \begin{pmatrix} 0 & \sigma_+^\mu \\ \sigma_-^\mu & 0 \end{pmatrix} \tag{2.46}$$

Where

$$\sigma_\pm \equiv (\mathbb{1}_{2\times 2}, \pm\vec{\sigma}) \tag{2.47}$$

And $\vec{\sigma}$ is a vector of the three Pauli matrices.

2.5.2 Spinors

For the u and v spinors we first define the helicity-eigenspinors χ_+ and χ_-

$$\chi_+(\mathbf{p}) = \frac{1}{\sqrt{2|\mathbf{p}|(|\mathbf{p}|+p_z)}} \begin{pmatrix} |\mathbf{p}|+p_z \\ p_x+ip_y \end{pmatrix} \tag{2.48}$$

$$\chi_-(\mathbf{p}) = \frac{1}{\sqrt{2|\mathbf{p}|(|\mathbf{p}|+p_z)}} \begin{pmatrix} -p_x+ip_y \\ |\mathbf{p}|+p_z \end{pmatrix} \tag{2.49}$$

These spinors satisfy

$$\frac{\vec{\sigma}\cdot\vec{p}}{|\mathbf{p}|}\chi_\lambda(\mathbf{p}) = \lambda\chi_\lambda(\mathbf{p}) \qquad \lambda = \pm 1 \tag{2.50}$$

In the limit where $p_y = 0$ and $p_x \to +0$

$$\chi_+(\mathbf{p}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.51}$$

$$\chi_-(\mathbf{p}) = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \tag{2.52}$$

The four spinors u and v are then defined as

$$u(p) = \begin{pmatrix} \omega_{-\lambda}(p)\chi_\lambda(\mathbf{p}) \\ \omega_\lambda(p)\chi_\lambda(\mathbf{p}) \end{pmatrix} \tag{2.53}$$

$$v(p) = \begin{pmatrix} -\lambda\omega_\lambda(p)\chi_{-\lambda}(\mathbf{p}) \\ \lambda\omega_{-\lambda}(p)\chi_{-\lambda}(\mathbf{p}) \end{pmatrix} \tag{2.54}$$

Where

$$\omega_\pm(p) \equiv \sqrt{E \pm |\mathbf{p}|} \tag{2.55}$$

2.5.3 Photon Polarization Vectors

For the photons helicity eigenvectors we first define two polarization vectors

$$\epsilon_1^\mu(p) = \frac{1}{|\mathbf{p}|p_T}\begin{pmatrix} 0 \\ p_x p_z \\ p_y p_z \\ -p_T^2 \end{pmatrix} \tag{2.56}$$

$$\epsilon_2^\mu(p) = \frac{1}{p_T}\begin{pmatrix} 0 \\ -p_y \\ p_x \\ 0 \end{pmatrix} \tag{2.57}$$

where

$$p_T = \sqrt{p_x^2 + p_y^2} \tag{2.58}$$

The helicity eigenvectors for helicities $\lambda = \pm 1$ are given by

$$\epsilon^\mu(p, \lambda = +1) = \frac{1}{\sqrt{2}}(-\epsilon_1^\mu(p) - i\epsilon_2^\mu(p)) \tag{2.59}$$

$$\epsilon^\mu(p, \lambda = -1) = \frac{1}{\sqrt{2}}(\epsilon_1^\mu(p) - i\epsilon_2^\mu(p)) \tag{2.60}$$

For $p_T \to 0$ we fix our polarization to

$$\epsilon_2^\mu(p_z) = \begin{pmatrix} 0 \\ p_z \\ 0 \\ 0 \end{pmatrix} \tag{2.61}$$

3 mAMEG Algorithm

3.1 Classes and Methods

An overview of the main `mAMEG` classes and methods:

1. `Process(process)`

   The `Process` class contains methods for creating contributing Feynman diagrams and matrix elements. The constructor takes a string as its argument which contains the process of interest. In `mAMEGv1.0` only $2 \to 2$ QED processes are

allowed. The notation for the input string is `"a b > c d"` where `a, b, c, d` are the incoming and outgoing particles of interest separated by `">"`. For example,

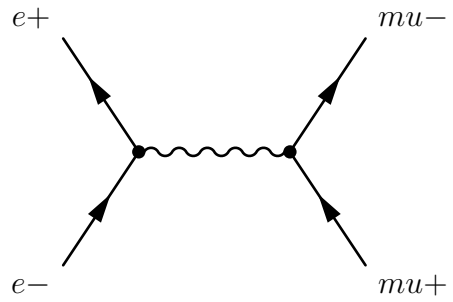
`"e+ e- > mu+ mu-"`

`"e- gamma > e- gamma"`

`"gamma gamma > e+ e-"`

(a) `.diagrams()`

The `diagrams` method creates and stores the contributing Feynman diagrams and matrix elements. Upon calling `diagrams()` a new directory within the current working directory is created where both the LaTeX script and Feynman diagram figures are stored. Along with creating and storing diagrams the `diagrams()` method returns a list of lists of strings which contain all of the contributing matrix element expressions of the form `[s_element, t_element, u_element]` where each element has a default value of `None`. For example in `"e- e+ > mu- mu+"` scattering the only contribution is from an s-channel diagram. The method creates and stores the following diagram



and returns a list

```
[[‘self.p3.wbar()’, ‘1j*self.dmu[mu]’, ‘self.p4.w()’, ‘-1j/s**2’,
 ‘self.p1.wbar()’, ‘1j*self.dml[mu]’, ‘self.p2.w()’], None, None]
```

2. `Particle(data, momentum, helicity)`

The `Particle` stores all of the particle data required to perform cross section calculations. The class is initialized with an instance of `ParticleData` taken from the `ParticleDatabase` (see Appendix A), the particle momentum, and

the particle helicity. The helicity has a default value of `None` which corresponds to unpolarized initial and final state particles.

(a) `.w()` and `.wbar()`

These methods compute the appropriate Dirac spinor used in the calculation of the matrix elements. See section two above or Appendix A.1 of [3] for details.

(b) `.e()`

This method computes the helicity eigenvectors for the external photon states used in the computation of matrix elements. Again, see section two above or Appendix A.2 of [3] for details.

3. `Collide(process, particle1, particle2, particle3, particle4)`

The Collide class implements the calculation of the differential cross-section. The constructor takes the $2 \to 2$ process of interest as a string and the particle data of the four particles in the form of instances of the ParticleDatabase class.

(a) `.me()`

The `.me()` method creates an instance of `Process(process).diagrams()` to generate the diagrams of the process and return the list of contributing matrix elements. The list of contributing matrix elements are multiplied and each amplitude is summed together with appropriate signs based upon the antisymmeterization of the matrix elements (see Rule 6 in section 2.4). The method returns a complex number representing the sum of all contributing transition amplitudes.

(b) `.xs()`

The `.xs()` method implements the calculation of the differential cross section. If the initial and final helicity states are specified by the user, the `.me()` method is called to obtain the transition amplitude of the process. The modulus squared of this value is taken to obtain the transition probability and plugged into our derived expression for the differential scattering cross section Eq.(2.37). If the initial and final particles are unpolarized (i.e. helicities are unspecified and take the default value of `None`) then the matrix element is calculated for each helicity configuration and the transition probability is **averaged** over the initial helicity states and **summed** over all final helicity states.

3.2 Using qtAMEG

Typical usage of `qtAMEG` is as follows

1. Create an instance of the `ParticleDatabase()` class

   ```
   pdb = ParticleDatabase()
   ```

2. Define momentum `FourVector` objects corresponding to your interested collision. Note that the momenta of particles three and four will be replaced in the calculation of the cross section.

   ```
   p1 = FourVector(-1, 0, 0, 40)
   ```

3. Create the four `Particle()` class instances

   ```
   particle1 = Particle(pdb["e-", p1, +1])
   ```

4. Create `Collide()` class instance with relevant process and matching particle info. Make sure that the order of the particles in the `process` string match the order of the input particle info.

   ```
   C = Collide("e- e+ > e- e+", particle1, particle2, particle3, particle4)
   ```

5. Initialize `Intregrator` class instance with `C.xs()` as an argument, along with interested solid angle

   ```
   i = Integrator(C.xs, 0, pi, 0, 2*pi)
   ```

6. Run the Monte-Carlo integration method within the `Integrator()` instance to obtain the scattering cross section for the solid angle of your choice in units of m^2.

   ```
   totxs = i.mc(1000)
   ```

4 Results

The table below shows the results obtained using `qtAMEG` for common tree-level QED processes. In order to check the accuracy of the computed unpolarized cross section I have used the invariance of the helicity averaged cross section under boosts. For each computed cross section, I have checked to see if the cross section remains the same after the original momenta are boosted along the z-direction by velocity β_z via the transformation

$$p \rightsquigarrow \Lambda_z p \tag{4.1}$$

With

$$\Lambda_z = \begin{pmatrix} \gamma & 0 & 0 & -\gamma\beta_z \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\gamma\beta_z & 0 & 0 & \gamma \end{pmatrix} \tag{4.2}$$

The results are as follows

Process	Total unpolarized σ [mb]	Boosted unpolarized total σ [mb]
$e + \mu \to e + \mu$	1.38×10^{-8}	1.34×10^{-8}
$e^- + e^+ \to e^- + e^+$	2.03×10^{-3}	3.02×10^{-3}
$e^- + e^+ \to \gamma + \gamma$	2.24×10^{-7}	2.54×10^{-7}
$e^- + e^- \to e^- + e^-$	2.05	1.08×10^{-1}
$e^- + \gamma \to e^- + \gamma$	4.83×10^{-5}	3.13×10^{-8}

Table 1: This tables shows the boosted and unboosted total scattering cross sections for common QED process calculated using `qtAMEG` at beam energies of 40 GeV and 1000 Monte-Carlo steps.

The results show that some of the cross sections are not invariant under the boost. This of course indicates that there is an error in the calculation and will need to be corrected in `qtAMEGv1.1`, I suspect that there is a combination of issues with the polarzation eigenstates and kinematics. Because `qtAMEG` only computes tree-level processes, a more thorough check would be to compute each cross section by hand and compare numerical results. To verify even further we could also compare to other automatic matrix element generators.

5 Conclusion

Given a QED process with appropriate syntax, `qtAMEG` successfully generates all contributing tree-level processes. Additionally, the program is able to correctly generate and output Feynman diagrams in a useful LaTeX format. We have also successfully produced boost-invariant spin-averaged total cross sections for a number of QED processes. Further work and debugging must be completed to achieve invariant scattering cross sections for all QED processes. In future versions of `qtAMEG` I would like to implement the weak vertex and improve upon the interface between user and program. Also, the runtime when averaging over photon polarizations is noticeably sluggish- approximately 10 minutes for 1000 MC points. This run time can be reduced by 1. Utilizing multiple cores and 2. Optimizing the python code. For future versions

I would like to parallelize the code and optimize for speed. The `qtAMEG` program is a stepping stone towards complete autonomous calculations of tree-level QED cross sections and even in its current form is a useful tool for generating contributing Feynman diagrams.

References

[1] Griffiths, David. Introduction to elementary particles. John Wiley & Sons, 2008.

[2] Raby, Stuart. Introduction to Field Theory and the Standard Model. In progress.

[3] H. Murayama, I. Watanabe and K. Hagiwara, "HELAS: HELicity amplitude subroutines for Feynman diagram evaluations," KEK-91-11.

[4] Gabrielse, G., et al. "New determination of the fine structure constant from the electron g value and QED." Physical Review Letters 97.3 (2006): 030802.

[5] Nogueira, Paulo. "Automatic Feynman graph generation." Journal of Computational Physics 105.2 (1993): 279-289.

[6] Hahn, Thomas. "Generating Feynman diagrams and amplitudes with FeynArts 3." Computer Physics Communications 140.3 (2001): 418-431.

[7] Aoyama, Tatsumi, et al. "Tenth-order QED contribution to the electron g-2 and an improved value of the fine structure constant." Physical Review Letters 109.11 (2012): 111807.

[8] Kilian, Wolfgang, Thorsten Ohl, and JŘrgen Reuter. "WHIZARD—simulating multi-particle processes at LHC and ILC." The European Physical Journal C 71.9 (2011): 1742.

[9] Alwall, Johan, et al. "MadGraph 5: going beyond." Journal of High Energy Physics 2011.6 (2011): 128.