Computer Physics Communications 323 (2026) 110069

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

A numerical approach to calculate cross sections for relativistic electrons and terrestrial gamma-ray flashes

H. van Gemert [a,*], F. Defranchi Bisso [a], C. Köhn [a,b]

[a] Department of Space Research and Technology, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark
[b] Department of Physics, Technical University of Dortmund, Dortmund, 44227, Germany

ARTICLE INFO

Prof. Dr. J Ballantyne

Keywords:
Cross sections
Quantum electrodynamics
Ab-initio calculation
Terrestrial gamma-ray flashes
Photodissociation
Particle modelling

ABSTRACT

Terrestrial gamma-ray flashes (TGFs) are bursts of energetic X- and gamma-rays which are emitted from thunderstorms as the Bremsstrahlung of relativistic electrons. While such relativistic particles propagate through the atmosphere, they interact with air molecules which can be studied with particle Monte Carlo models. Such models require cross sections as an input. Whilst there are well established data for elastic scattering, Bremsstrahlung and impact ionization through relativistic electrons as well as for photoionization, Compton scattering and pair production for energetic photons, we lack cross sections for photoexcitation, photodissociation or the excitation of air molecules through electrons which can contribute to the chemical activation of the atmosphere, potentially relevant for the study of greenhouse gas production from lightning in our atmosphere. In order to fill this cross section gap, we here present a novel numerical tool calculating cross sections directly from Feynman diagrams providing both differential and total cross sections. We provide an overview of the code structure and present benchmarking cases against well-known cross sections. Additionally, we will present a first application by calculating the cross section for photodissociation for a wide range of energies. The presented model is capable of calculating cross sections for more leptonic and photonic processes in the atmosphere than today. Subsequently, this will allow for more realistic simulations of energetic phenomena in our atmosphere.

Program summary

- Program title: Quantum Pathways: Feynman's Numerical Solver (QP-FNS)
- CPC Library link to program files: https://doi.org/10.17632/ggxh3wv2by.1
- Developer's repository link: https://gitlab.gbar.dtu.dk/chrstk/cross-section-calculation
- Licensing provisions: GNU General Public License 3
- Programming language: C++
- Nature of problem: This program calculates a numerical dataset for the total cross section and singly differentiated cross section (with respect to scattering angle), based on the Feynman diagrams for 2→2 particle processes. The cross sections can be calculated in either the laboratory or center of mass reference frame.
- Solution method: The program uses quantum field theory to translate Feynman diagrams into quantifiable expressions, which it discretizes to calculate the scattering matrix. This is combined with expressions for the problem's 4-momenta to solve for the differential cross section. Numerically integration is used to obtain the total cross section.
- Additional comments: Energy input and output is in units of MeV. The Feynman diagram input is restricted to 2→2 tree level diagrams. Input expressions are in natural units, output expressions are in either natural or SI units, depending on which is requested in the input. See appendix A on how to compile the program, and appendix B on how to input diagrams.

* Corresponding author.
E-mail addresses: hannahvangemert@gmail.com (H. van Gemert), defranchi.fede@gmail.com (F. Defranchi Bisso), koehn@space.dtu.dk (C. Köhn).

https://doi.org/10.1016/j.cpc.2026.110069
Received 9 April 2025; Received in revised form 19 December 2025; Accepted 3 February 2026
Available online 11 February 2026

1. Introduction

Terrestrial gamma ray flashes (TGFs) are bursts of high-energy photons with quantum energies of up to 40 MeV, lasting for tens to hundreds of microseconds [1–6], which are produced as the Bremsstrahlung of relativistic electrons [7,8]. However, when producing Bremsstrahlung photons, it is more likely that only a small friction of the incident electron energy is transferred towards the emitted Bremsstrahlung photon [8] such that a typical beam will also contain low-energy photons. Previous studies [9–11] indicate that a typical TGF energy distribution can be approximated by $\sim \exp(-E_\gamma/E_c)/E_\gamma$ where E_γ is the photon energy and E_c the exponential fall-off energy extending photon energies from MeV energies down to some tens of eV. Relativistic electrons and photons travel through the atmosphere, they interact with the gas molecules of our atmosphere up to satellite altitudes. As a supplement to flight [12] and space measurements [13–17], a widely used approach to study these energetic particle beams are numerical simulations, typically particle simulations where electrons and photons are treated as superparticles [18–20]. A main component of such simulation tools, are the cross sections of collision processes determining the probability of an electron or a photon scattering off an air molecule. Cross section data are available for the elastic scattering [21,22], Bremsstrahlung [7,8] and impact ionization of relativistic electrons [23–25] as well as for photoionization, Compton scattering, pair production and hadron production through photons [7,8,26–28] producing new particle species such as positrons, protons and neutrons in our atmosphere [20,29]. Since the accuracy of cross section determines the quality of simulation results, there is a constant need to check and improve cross section data; for example, in recent work, Schmalzried and Luque [22] showed how small variations in elastic scattering cross sections can significantly influence simulations of run-away electron generation, and provided a new set of cross section data for the High-Energy Atmospheric Electricity community. Energetic electrons and photons might be capable of altering the chemical composition of the atmosphere [30,31], leading to a chemically more active environment which might play a significant role in the production of greenhouse gases or in the production of organic molecules, e.g. in the atmosphere of Saturn's moon Titan [32]. However, to study such phenomena, additional processes are needed such as the excitation of air molecules through relativistic electrons (or positrons) as well as photoexcitation or photodissociation whose cross sections are not or only hardly known. For example, cross sections for photodissociation are only available for certain molecules and limited energy ranges [33].

Cross sections can be obtained through experiments, numerical modelling or analytic calculations. Whilst there is only limited data available through experiments and numerical modelling, analytic calculations are typically tedious and complicated. Examples of such analytical calculations are shown by [7] and [8]. Aside from being complicated and time consuming, such calculations face another issue: They often involve approximations, prohibiting exact results, see a discussion by [28]. Some cross section databases [34] are more extensive, but they are still limited in the availability of certain cross sections. Cross sections used by [35] and those from [36] are obtained by directly fitting experimental data. This method is not necessarily limited algebraically. However, it only works if the measurement is provided in good enough detail, as to lower the uncertainty of the interpolation between the data points. It can also be quite intensive to capture data this way for different gases and energy ranges. In any case, obtaining cross sections through experiments is an extensive undertaking as a new set of measurements needs to be performed for each individual scattering process.

The issues above can be partially circumvented by automating the calculation of cross sections with a numerical method. Numerical calculations are typically faster than the theoretical calculation and they are more easily accessible than the experimental approach. Many such numerical schemes have been proposed over the years, as summarized by [37]. Some focus on performing part of the cross section calculation. They for example only perform the phase space integration, or take care of Feynman diagram generation like [38]. Others are more comprehensive in that they can evaluate either a particular type of diagram, or even different types to calculate the scattering matrix. Programs like CompHep ([39]) or the FeynCalc implementation for Mathematica ([40]) even perform the entire calculation, starting from the QED Lagrangian for particular scattering processes. The latter is very well known for its symbolic computational power. Packages like FeynCalc are a useful tool to get the cross section expressions, but are not necessarily meant to calculate numerical data sets with. CompHep does provide actually cross section data, but is limited to high energies only. Furthermore, CompHep is a relatively old program, like many others mentioned by Harlander and Steinhauser [37]. We therefore suggest a program more suited directly for calculating the different cross section datasets for atmospheric processes.

That is why here we present a numerical tool of calculating cross sections for Quantum Electrodynamics (QED) processes for leptons and photons, such as photodissociation, allowing to make atmospheric electricity particle simulations far more realistic in the future. Like with the analytical method, we start with Feynman diagrams to directly translate the process into a mathematical expression. However, the presented program does this translation in an automated fashion and calculates the resultant differential and total cross section numerically for a wide energy range, covering low and high energies. The program presented here is designed and tested on QED tree-diagrams, assuming two initial and two final state particles, as relevant for most collision processes in the atmosphere.

The notation throughout this paper is consistent with the one used by Peskin and Schroeder [28]. The main points are:

- Natural units are used unless specified otherwise, i.e. $\hbar = c = 1$.
- The Einstein notation is used, unless specified otherwise.
- Boldfaced quantities (e.g. $\mathbf{p}$) denote a 3-vector. Matrices and 4-vectors are denoted in italics.
- The Minkowski spacetime metric $g_{\mu\nu}$ is used with $(+---)$ convention.
- The γ-matrices γ^μ are defined in *Weyl/Chiral* representation.
- The slashed notation on vectors stands for $\not{p} = \gamma^\mu p_\mu$.

In Section 2, we first show how to obtain cross sections in different reference frames and how to translate Feynman diagrams in a discretized manner before presenting the code setup and structure. Benchmarking is presented in Section 3 demonstrating the program's functionality and reliability. In Section 4 we demonstrate a first application of the code by calculating the cross sections for the photodissociation of molecular nitrogen. In Section 5 we conclude on the code's performance and resulting cross sections and provide suggestions for future development. Details on how to manipulate, execute and run the code are given in the appendix.

2. Theory and numerical implementation

Before discussing the numerical implementation of the cross section calculation in Section 2.3, let us review briefly in Sections 2.1 and 2.2 how differential and total cross sections are calculated.

2.1. Cross sections

The differential cross section (DCS) of two initial particles $\mathcal{A}$ and $\mathcal{B}$ scattering with each other and producing n_f final particles is calculated by combining the initial state of $\mathcal{A}$ and $\mathcal{B}$, the probability of these two particles scattering into n_f final particles, and the number densities of the initial system. According to [28], this leads to the following expression for the differential element $d\sigma$:

$$d\sigma = \frac{1}{2E_\mathcal{A} 2E_\mathcal{B} |v_\mathcal{A} - v_\mathcal{B}|} |\mathcal{M}|^2 \times \prod_f \frac{d^3 p_f}{(2\pi)^3} \frac{1}{2E_f} (2\pi)^4 \delta^{(4)}\left(p_\mathcal{A} + p_\mathcal{B} - \sum p_f\right).$$

(1)

Here $v_{\mathcal{A},\mathcal{B}} = p^3_{\mathcal{A},\mathcal{B}}/p^0_{\mathcal{A},\mathcal{B}}$ is the relativistic velocity (without loss of generality oriented in z-direction), and $E_{\mathcal{A},\mathcal{B}} = p^0_{\mathcal{A},\mathcal{B}}$ are the total particle energies. p_f and E_f are the momenta and the energies of the particles in the final state, respectively. $\mathcal{M}$ is the process' scattering matrix which is defined as the product of all components in the process' Feynman diagram [28,41], as we show more explicitly in paragraph 2.2.

To obtain the expression for the DCS, the multidimensional differentials over the final particles' momenta in Eq. (1) need to be re-expressed.

In the case of two final particles, the differentials can be evaluated directly, depending on the frame of reference, leading to the following in the center-of-mass frame (CM) [28]:

$$\left(\frac{d\sigma}{d\Omega}\right)_{CM} = \frac{1}{2E_{\mathcal{A}}2E_{\mathcal{B}}|v_{\mathcal{A}} - v_{\mathcal{B}}|}\frac{|\mathbf{p}_1|}{(2\pi)^2 4(E_{\mathcal{A}} + E_{\mathcal{B}})}|\mathcal{M}|^2. \tag{2}$$

where $E_{\mathcal{A}} + E_{\mathcal{B}}$ is the CM energy of the initial particles, $d\Omega = d\phi\sin(\theta)d\theta$ is the solid angle element and $|\mathbf{p}_1|$ is the 3-norm of the spatial part of either one of the final state particles.

In the laboratory frame, we can reduce the momentum differentials as follows:

$$\begin{aligned}\frac{d^3p_1}{(2\pi)^3}\frac{d^3p_2}{(2\pi)^3}(2\pi)^4\delta^{(4)}(p_{\mathcal{A}} + p_{\mathcal{B}} - p_1 - p_2) &= \frac{d^3p_1}{(2\pi)^3}(2\pi)\delta(E_{\mathcal{A}} + E_{\mathcal{B}} - E_1 - E_2)\\ &= \frac{d^3p_1}{(2\pi)^3}(2\pi)\delta\Big(E_{\mathcal{A}} + E_{\mathcal{B}} - \sqrt{m_1^2 + |\mathbf{p}_1|^2}\\ &\quad - \sqrt{m_2^2 + |\mathbf{p}_2|^2}\Big)\end{aligned}$$

where we used the spatial part of the delta function in the first step and the mass-shell relation between particle momenta and energies in the second. Note that the mass-shell relation still holds for photons where the rest mass m becomes 0. Transforming the remaining differential element to spherical coordinates and using the properties of the delta function with respect to $d|\mathbf{p}_1|$ yields

$$\frac{d\Omega}{(2\pi)^2}|\mathbf{p}_1|^2\left|\frac{|\mathbf{p}_1|}{E_1} + \frac{|\mathbf{p}_2|}{E_2}\frac{d|\mathbf{p}_2|}{d|\mathbf{p}_1|}\right|. \tag{3}$$

Finally, we assume that $\mathcal{B}$ is stationary whilst $\mathcal{A}$ is moving, and define $\theta = \sphericalangle(\mathbf{p}_{\mathcal{A}}, \mathbf{p}_1)$ to be the angle between the initial and final state of particle $\mathcal{A}$, thus

$$\frac{d^3p_1}{(2\pi)^3}\frac{d^3p_2}{(2\pi)^3}(2\pi)^4\delta^{(4)}(p_{\mathcal{A}} + p_{\mathcal{B}} - p_1 - p_2) = |\mathbf{p}_1|^2\left|\frac{|\mathbf{p}_1|}{E_1} + \frac{|\mathbf{p}_1| - |\mathbf{p}_{\mathcal{A}}|\cos\theta}{E_2}\right|\frac{1}{(2\pi)^2}d\Omega.$$

Combining Eqs. (4) and (1), we obtain the DCS in the laboratory reference frame:

$$\left(\frac{d\sigma}{d\Omega}\right)_{lab} = \frac{1}{2E_{\mathcal{A}}2E_{\mathcal{B}}|v_{\mathcal{A}} - v_{\mathcal{B}}|}|\mathbf{p}_1|^2\left|\frac{|\mathbf{p}_1|}{E_1} + \frac{|\mathbf{p}_1| - |\mathbf{p}_{\mathcal{A}}|\cos\theta}{E_2}\right| \times \frac{1}{4E_1E_2}\frac{1}{(2\pi)^2}|\mathcal{M}|^2, \tag{4}$$

Note that in both both Eqs. (2) and (4), the actual physics of the scattering process is encoded in $\mathcal{M}$, whose numerical calculation is discussed in the next two subsections.

To obtain the total cross section (TCS) σ, we integrate over the solid angle Ω:

$$\sigma = \int \frac{d\sigma}{d\Omega}d\Omega = 2\pi\int_0^{\theta_{max}}\frac{d\sigma}{d\Omega}\sin(\theta)d\theta \tag{5}$$

where we use that the DCS does usually not depend on ϕ.

Before calculating (5), we need to check the limits of θ. If all final particles are distinguishable, $\theta_{\max} = \pi$. But if some final state particles are identical, we divide $\theta_{\max}$ by the number of identical particles. For two final particles, it is $\theta_{\max} = \pi/2$. We also expect the DCS to be symmetric in this situation, in which case it does not matter if we integrate over $0 \le \theta \le \pi/2$ or $\pi/2 \le \theta \le \pi$.

2.2. The scattering matrix $\mathcal{M}$

Let us now briefly recall elements of the scattering matrix $\mathcal{M}$ which needs to be discretized for numerical cross section calculations. In short, $\mathcal{M}$ is the analytic equation of a Feynman diagram, which draws the particle interaction process according to specific rules. Such diagrams consist of several components. The components of the first-order tree diagrams considered here can be expressed according to Yukawa theory and the rules of QED [41,42]. The relation between the diagram components and their mathematical expressions are shown in Table 1. In our case, external particle lines represent the initial and final state particles, vertices are the interaction points, and propagators connect one vertex with the next.

Table 1
List of the different Feynman diagram components in Yukawa theory and QED. Left: component drawing. Middle: component mathematical expression. p here stands for the component's 4-momentum vector, m for the mass (in natural units), μ and ν are vertex indices from $0-3$, and s is an index for polarization or spin, from $0-1$. Right: Component's reference number and name.

μ ~ p ~ ν	$\frac{-ig_{\mu\nu}}{p^2}$	1. Photon Propagator
μ — p — ν	$\frac{i(\not{p}+m)}{p^2-m^2}$	2. Fermion Propagator
μ	$-ie\gamma^\mu$	3. Photon Vertex
μ	$-ig$	4. Fermion Vertex
p	$u^s(p)$	5. External Fermion, *entering vertex*
p	$\bar{u}^s(p)$	6. External Fermion, *exiting vertex*
$\bar{p}$	$\bar{v}^s(p)$	7. External Anti-Fermion, *entering vertex*
$\bar{p}$	$v^s(p)$	8. External Anti-Fermion, *exiting vertex*
p	$\epsilon^s_\mu(p)$	9. External Photon, *entering vertex*
p	$(\epsilon^s_\mu(p))^*$	10. External Photon, *exiting vertex*

The external fermion components $u^s(p)$ are spinors in the plane wave solution

$$\psi(x) = u(p)e^{-ip\cdot x} \tag{6}$$

of the Dirac equation (similar for $v^s(p)$) where s refers to the spin state of the particle. $u^s(p)$ relates to positive frequency solutions (fermions), whereas $v^s(p)$ relates to negative frequency solutions (anti-fermions). $\bar{u}^s(p)$ and $\bar{v}^s(p)$ denote conjugate spinors. Explicit expressions for these spinors are [28]:

$$u^s(p) = \begin{pmatrix}\sqrt{p\cdot\sigma}\xi^s\\ \sqrt{p\cdot\bar{\sigma}}\xi^s\end{pmatrix},\ s = 0, 1. \tag{7}$$

$$\bar{u}(p) = u^\dagger(p)\gamma^0, \tag{8}$$

$$v^s(k) = \begin{pmatrix}\sqrt{p\cdot\sigma}\eta^s\\ -\sqrt{p\cdot\bar{\sigma}}\eta^s\end{pmatrix},\ s = 0, 1. \tag{9}$$

$$\bar{v}(p) = v^\dagger(p)\gamma^0 \tag{10}$$

where $\sigma = (\sigma^0, \sigma^1, \sigma^2, \sigma^3)$ is the four-vector of Pauli matrices and ξ as well as η are chosen such that $\xi^\dagger\xi = 1$ (likewise for η).

Feynman diagrams implicitly consist of one temporal and one spatial dimension identifying the particles' motion, where the temporal axis of a diagram is considered to be along the direction of the propagator in a $2 \to 2$ particle process. The time-ordered multiplication of all Feynman diagram components leads to the scattering matrix $i\mathcal{M}$ as it appears in Eq. (1). This means that $i\mathcal{M}$ begins with the vertex (and external particles) at the latest time. After this vertex follows the propagator entering the second latest vertex, etc. A particle interaction process can consist of multiple diagrams; in such a case, a summation of all diagrams needs to be computed. The sign of each diagram in the summation depends on the amount of fermion exchanges with respect to the first diagram.

An odd number of fermion exchanges results in a minus sign, an even number in a plus sign.

The total calculation for $\mathcal{M}$ follows as

$$i\mathcal{M} = \sum_{k=1}^{N_{diag}} (-1)^{N_{exchange,k}} \prod_{j=1}^{N_{comp}} C_{j,k}, \tag{11}$$

where j labels the total number of N_{comp} different components $C_{j,k}$ of a diagram D_k (summarized in Table 1) that need to be multiplied. The sum is taken over all diagrams of a process where $N_{exchange,k}$ stands for the number of fermion exchanges of the considered diagram k with respect to the first lowest-order Feynman diagram. Some of the component expressions in Table 1 used as an input to Eq. (11) show index dependencies because of the four-dimensionality of space-time. Let us first consider the vertices themselves: Photon vertices carry a 4-vector index μ. Together with either a photon propagator (with two indices $\mu\nu$) or an external photon (with one index μ), this results in an Einstein summation when multiplied in Eq. (11). Each summation pair over two recurring indices is in principle independent off that summation index, which remains the case within a single diagram. However, it can be shown that by pulling these Einstein summations out of the diagram summation, the indices between diagrams can become linked. This reduces the number of independent indices to the number of vertices in one diagram instead of the total number of vertices. Although this does not reduce the total number of iterations, it does provide a computational convenience, as shown more explicitly in paragraph Section 2.3.2. Particularly, for two processes with two initial and two final state particles, it is $N_{vertex} = 2$.

Calculating cross sections, we are interested in the absolute value squared of the scattering matrix. For a particular set of particles in a particular state, this is simply $|\mathcal{M}|^2 = i\mathcal{M} \cdot (i\mathcal{M})^*$. Particular states of the particles refer to the different spin vectors ξ or η of the fermion spinors, see Eqs. (7–10) and the different photon polarizations ϵ which can be represented by two orthonormal basis vectors. These vectors can be freely chosen in any desired form, as long as they are orthonormal to the direction of motion of the photon. For an external photon with 4-momentum $k^\mu = \begin{pmatrix} k & 0 & 0 & k \end{pmatrix}$ a generic choice of polarization vectors can be $\epsilon^0_\mu = \begin{pmatrix} 0 & 1 & i & 0 \end{pmatrix}$ and $\epsilon^1_\mu = \begin{pmatrix} 0 & 1 & -i & 0 \end{pmatrix}$. for polarizations $s = 0$ and $s = 1$. For photons not along the z-axis, the polarization vectors can be calculated through $\epsilon^0_\mu = k \times z$ and $\epsilon^1_\mu = (k \times z) \times k$ where $z = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}$ represents the z-axis and where the spatial parts of resulting polarization vectors have been normalized. Both spin and polarization indices are independent of other Feynman components.

Although individual particles have distinct spin or polarization, physical measurements will typical be in the context of several particles with not necessarily the same polarization or spin states. Hence, we are not interested in particular polarizations or spin states and remove their dependency by averaging over the states or polarizations of the initial particles, and summing over the spin/polarization states of the final particles. For $2 \to 2$ particle processes, this results in a factor $1/4$ for the scattering matrix and four more summations. In such summations we refer to the spin states or polarizations of the initial particles (depending on whether they are fermions or photons) as s, s' and use r, r' for final state particles.

By combining Eq. (11) with the above, the scattering element of the cross section becomes

$$|\mathcal{M}|^2 = \frac{1}{4} \sum_s \sum_{s'} \sum_r \sum_{r'} i\mathcal{M} \cdot (i\mathcal{M})^* = \frac{1}{4} \sum_{s,s',r,r'} \left(\left[\sum_{\mu,\nu} \sum_k (-1)^{N_{exchange,k}} \prod_j C_{j,k} \right] \left[\sum_{\mu,\nu} \sum_k (-1)^{N_{exchange,k}} \prod_j C_{j,k} \right]^* \right). \tag{12}$$

2.3. Program structure

The cross section calculations outlined in Sections 2.1 and 2.2 can be automated resulting in a numerical tool whose structure is outlined in Fig. 1. The program consists of three different phases: receiving user input, calculating $\mathcal{M}$, and calculating the DCS and TCS. These phases are displayed by dotted rectangles in Fig. 1 and are described in more detail in the following subsections. Instructions how to compile as well as run the code, and details about the output are found in Appendix A. The code is available as Supplementary Material and through `https://gitlab.gbar.dtu.dk/chrstk/cross-section-calculation.git`.

2.3.1. Input processing

All user input is done in a single header file (an overview of the general structure is given in Appendix B; an example is given in Appendix C). Feynman diagrams are provided based on their vertices. A diagram is implemented by adding all vertices from earliest to latest (see examples in the folder `input files` for the cases presented in Sections 3 and 4). For each vertex it is specified which external particles are connected to it and what propagator connects it to the next vertex. In the case of $2 \to 2$ particles in tree-diagrams, each diagram has two vertices. Particles and propagators are constructed as objects in the code by specifying their type, direction, momentum, and mass. Types can be fermion (1), antifermion (-1), or photon (0). Direction refers to the particle's momentum vector entering or exiting the connected vertex (depending on time and fermion/antifermion state). Momenta are 4-vectors and input as functions of the energy E depending on the respective case and scattering angle θ. External particle objects can - and should - be reused in multiple vertex creations, since these represent the same particle. Propagators and vertices, however, are not the same between diagrams and need to be created anew for each instance. We refer to the collection of a vertex with its connected external particles and propagator as a vertex group (see Fig. 2).

The energy range E is free for the user to specify. The range of E as well as the angle θ as defined in Eq. (4) and their accuracy are user specified as part of the general program input where θ should always be calculated from 0 to π. However, the code allows for other ranges of θ, or even single angular values; this will produce values for the differential cross section, but will not calculate the total cross section which is the integration of DCS over Ω (Eq. (5)). Furthermore, the user can specify if the considered process is in the center-of-mass frame or the laboratory frame, and if the CS values should be given in natural units or SI units. Note that all input is always done in units of MeV, regardless of the output. The program has predefined mass variables for photons, electrons/positrons, protons, neutrons, muons and tauons.

A thorough description of how input files look like and can be structured is given in Appendix B.

2.3.2. Calculating $|\mathcal{M}|^2$

To calculate the scattering matrix, we employ Eq. (12). Due to the matrix and vector nature of most Feynman components, the order in the product $\prod_j C_{j,k}$ matters. Each vertex group is ordered based on its algebraic properties, which is demonstrated in Fig. 2. Vertices are always matrices, and external fermions are always vectors. Thus, the fermions get placed directly around the vertex. If the group has a propagator, it will be either a matrix or a scalar, and in either case, can be multiplied first in the group. External photons are scalars, and therefore do not necessarily have a designated position. For consistency, we define them to always appear last in the group. Component ordering is saved as an array. Vertex groups themselves should have time ordering, i.e. the input order is reversed, hence vertices go from latest to earliest. Note that vectors and scalars are cast into the form of a complex 4x4 matrix to ease numerical implementation. Column vectors are implemented as the first column of a matrix, with the rest of the entries set to zero; similarly, row vectors as the first row, and scalars as the top left entry. This maintains the components algebraic properties while avoiding type and function errors in code because of different dimensionality. Ordering is done for each diagram in the process, such that we end up with an ordered array of ordered vertex groups for each diagram.

After marking which components in the code have indices s, s', r, r' for spin and polarization averaging and which Einstein summations over

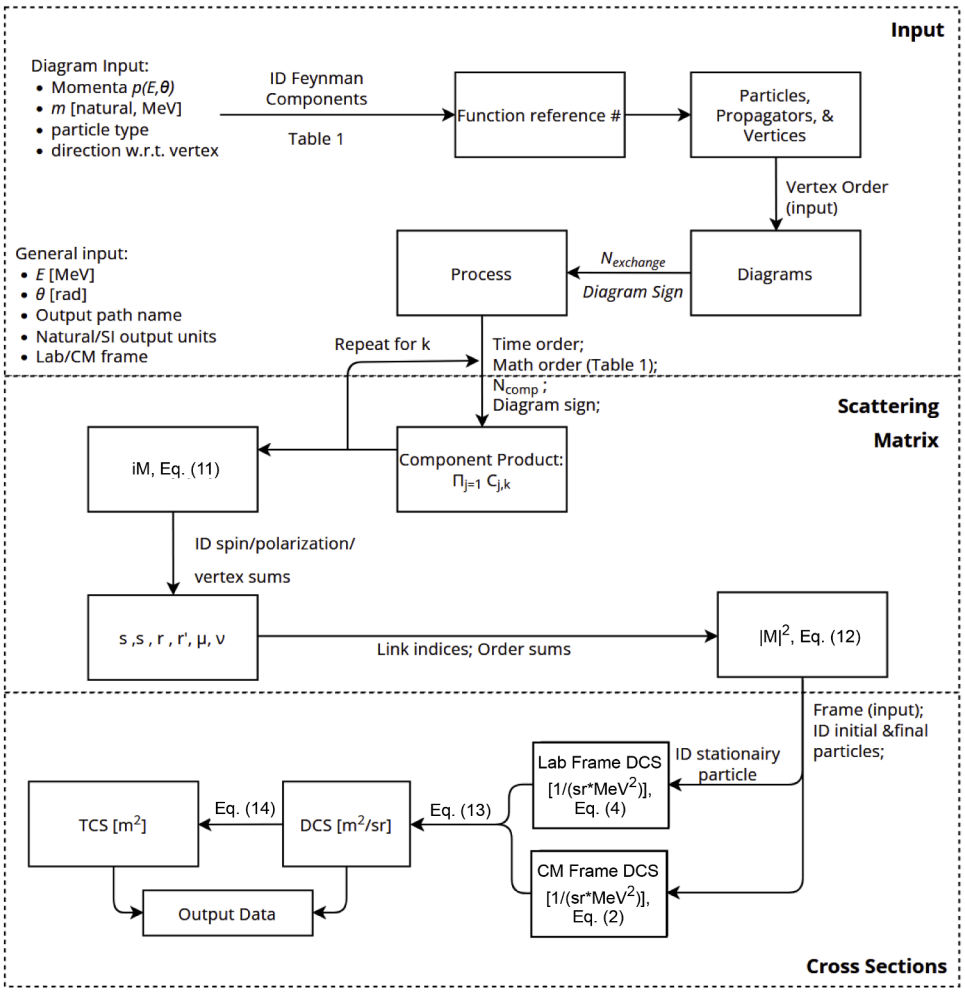


Fig. 1. Flow diagram of the program. The three dotted, rectangles indicate three phases. From top to bottom: user input, calculation of $|\mathcal{M}|^2$ (M in figure), and calculation of the DCS and TCS. The diagram starts at "Diagram Input", and ends at "Output Data".

Fig. 2. Order of a single vertex group. External particles are indicated in grey, the vertex in white, and the propagator in dotted white. There are only two external particles in a vertex group for $2 \to 2$ particle processes. The four shown here are only to indicate the ordering. The propagator is optional, since the latest vertex in the diagram does not have a propagator in its group.

μ and ν need to be performed, $|\mathcal{M}|^2$ is calculated for a particular set (E, θ) according to Eq. (12).

2.3.3. DCS and TCS

Once $|\mathcal{M}|^2$ is obtained, the code proceeds to calculate the DCS depending on the frame of reference of the collision, specified as user input. A program warning is given if the implemented process is not a $2 \to 2$ process, which might indicate an error in the input.

- For the **laboratory frame**, the program uses Eq. (4) to calculate the DCS. We identify $p_{\mathcal{A}}$ by checking which initial particle does not have zero spatial momentum components. p_1 is chosen such that this is the momentum of the particle that encloses the angle θ with the direction of the initial particle $\mathcal{A}$. If both particles have the same angle, the final particle that was inputted first, is used. The momenta of all particles are then plugged into Eq. (4), yielding the final DCS.
- For the **center-of-mass frame**, Eq. (2) is used. This value is directly calculated once the initial and final state particles are identified. Note that for a system with 2 initial particles, their energies are equal.

The DCS is by default calculated in natural units. If the user specifies the output to be in SI units, the conversion is performed through

$$DCS_{\text{SI}}[\text{m}^2] = DCS_{\text{natural}}[1/\text{MeV}^2] \cdot (\hbar c)^2 \tag{13}$$

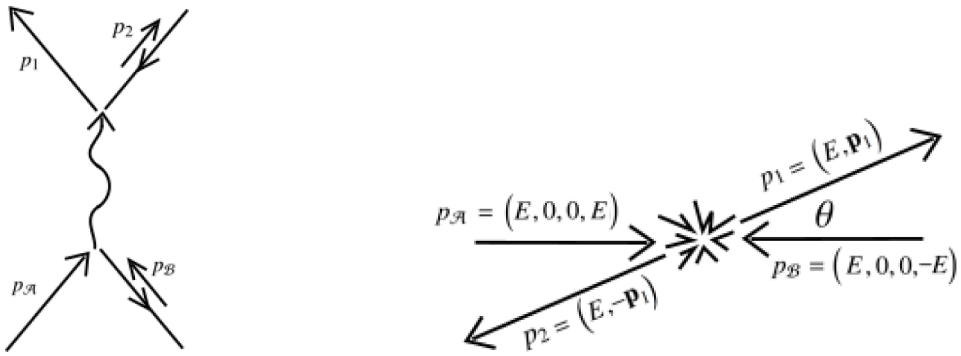
where $\hbar c = 197.327 \cdot 10^{-15}$ MeV · m.

The TCS is calculated by numerically integrating the DCS (5) as

$$\sigma = 2\pi \sum_{j=0}^{j_{max}} \frac{d\sigma}{d\Omega} sin(\theta_j) \Delta\theta \tag{14}$$

where the number of integration points $j_{\max}$ is chosen such $\theta_{j_{\max}} = \theta_{\max}$ and where $\Delta\theta = \theta_{\max}/j_{\max}$ determines the integration step size.

As mentioned in Section 2.1, $\theta_{\max}$ is different depending on whether the two particles in the final state are identical, or not. This is determined by checking if the particles' mass and type are the same. However, the user can overwrite this check through the input file, in which case the program will not consider the final particles identical, regardless of its own check. This becomes relevant for processes that contain final state fermions of the same mass, but that are not identical from a QED



(a) (b)

Fig. 3. (a) First order Feynman diagram for $e^+e^- \to \mu^+\mu^-$. Here $\mathcal{A}$ is the electron, $\mathcal{B}$ is the positron, particle 1 is μ^+ and particle 2 is μ^-. (b) Collision in the center of mass frame for a $2 \to 2$ particle process, with general expressions for the 4-momenta p, where $\mathbf{p}_1 \cdot \hat{\mathbf{e}}_z = |\mathbf{p}_1| \cos\theta$.

point of view. We will see this case applied for the photodissociation cross sections presented in Section 4.

If any DCS values are infinite, they are excluded from the TCS integral by setting them to zero beforehand. This is to avoid integration issues and to be able to get a fully numerical output dataset. In such cases, the program will produce a warning with the number of points that was set to zero. Such manual zero points are typically clearly recognizable in a data set, since they show as abrupt cutoffs. They are often due to the input energy range including energies smaller than the system rest energy, in which case the DCS can be considered to be zero. However, we advise care with the zero values in other parts of the output DCS. If they occur due to numerical issues, it is still reasonable to use the TCS; if infinities or invalid values occur due to physics, we advise to not use the provided TCS data for the energies in question, nor to use the DCS artificial zero points.

3. Benchmarking

To evaluate the accuracy of the program implementation, we use four general particle processes with known analytical solutions - hereafter called the "reference" - to test the numerical results against. The processes used for this comparison are $e^+e^- \to \mu^+\mu^-$, Bhabha scattering, pair annihilation, and Compton scattering, and are presented in the following paragraphs. The reference for the TCS is acquired by numerically integrating the reference DCS using the known *Python* module *scipy.integrate.quad* ([43]). The Feynman diagrams for benchmarking are drawn based on [28], unless specified otherwise. For all processes the relative difference between our numerical results and the references are calculated as

$$\Delta\sigma = \frac{\sigma_{\text{numerical results}} - \sigma_{\text{reference}}}{\sigma_{\text{reference}}}. \tag{15}$$

In order to calculate the processes, you need to know the Feynman diagram itself, as well as the momenta in the considered reference frame in terms of E and θ. The input files for these cases are provided in the folder `input files` in the main code directory, see Appendices B and C for details.

3.1. Process $e^+e^- \to \mu^+\mu^-$

The first order Feynman diagram for this process is shown in Fig. 3a. We use this process to test the program's handling of external fermions, photon propagators, and generally the DCS and TCS calculation.

The process is calculated in the center of mass frame, see Fig. 3b. We define θ to be the angle between μ^- and e^-, with e^- moving along the z axis. According to [28], the reference exact analytical expression for the DCS in this situation is

$$\frac{d\sigma}{d\Omega} = \frac{\alpha^2}{4E_{CM}^2}\sqrt{1-\frac{m_\mu^2}{E^2}}\left[\left(1+\frac{m_\mu^2}{E^2}\right)+\left(1-\frac{m_\mu^2}{E^2}\right)\cos^2\theta\right]. \tag{16}$$

The example input file for this process is listed in Appendix C.1. The numerical results are shown in Figs. 4 and 5 and compared with Eq. (16). All DCS panels are shown for energies above the rest energy of a muon (105.6 MeV), except the far left, which correctly shows a zero DCS at lower energy. There is hence no relative difference plot for this energy. There is an excellent agreement between the numerical results and the reference, within a few sub-percent for both the DCS and the TCS. The energy E can be defined here as being the energy of any of the particles since this collision process is considered in the center-of-mass frame, see Fig. 3b.

3.2. Bhabha scattering: $e^+e^- \to e^+e^-$

The first order Feynman diagram for this process is shown in Fig. 6. We use this process to test the program's handling of multi-diagram calculations with a relative sign difference.

The process is calculated in the center-of-mass frame. The input file for this specific scattering case is given in Appendix C.2. We define θ to be the angle between the two electrons, with the initial e^- moving along the z-axis. According to [28], the exact analytical expression for the DCS for Bhabha scattering in the center-of-mass frame is

$$\frac{d\sigma}{d\Omega} = \frac{1}{2\pi}\frac{\pi\alpha^2}{s}\left(u^2\left(\frac{1}{s}+\frac{1}{t}\right)^2+\left(\frac{t}{s}\right)^2+\left(\frac{s}{t}\right)^2\right) \tag{17}$$

where the Mandelstam variables s, t, u are defined as

$$\begin{aligned} s &= (p_{\mathcal{A}} + p_{\mathcal{B}})^2 \\ t &= (p_{\mathcal{A}} - p_1)^2 \\ u &= (p_{\mathcal{A}} - p_2)^2 \end{aligned} \tag{18}$$

Numerical and analytic results are shown in Figs. 7 and 8 for various energies with an excellent agreement between these two, having the relative difference within a few sub-percent for both DCS and TCS. The energy E can be defined here as being the energy of any of the particles since this collision process is considered in the center-of-mass frame.

3.3. Pair annihilation: $e^+e^- \to 2\gamma$

The first order Feynman diagram for this process is shown in Fig. 9. We use this process to test the program's handling of multi-diagram calculations, external photons, and identical final particles.

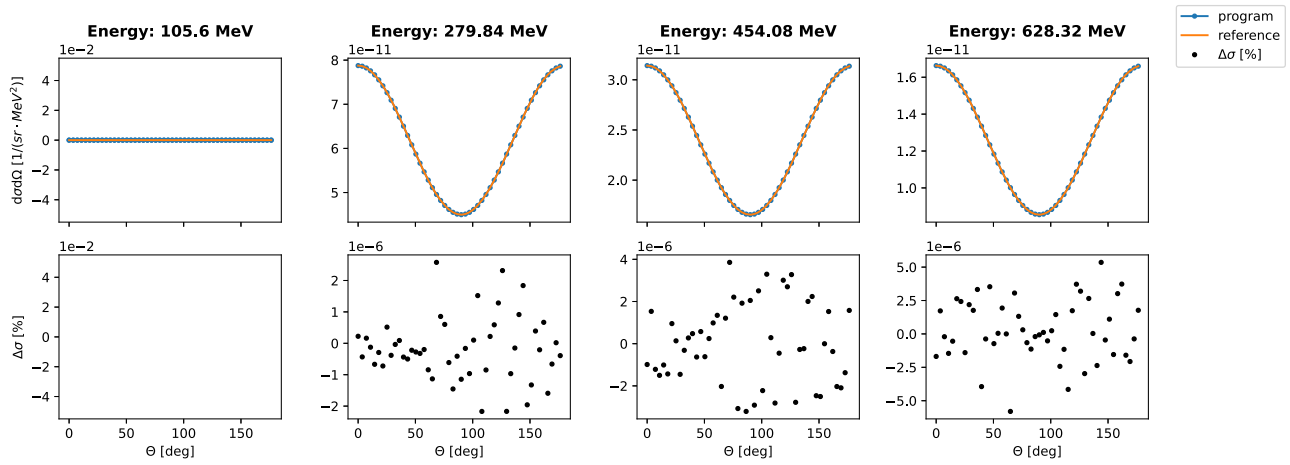


Fig. 4. Top: DCS $d\sigma/d\Omega$ (Eq. (16)) for $e^+e^- \to \mu^+\mu^-$ in the center-of-mass frame in natural units. The analytic cross sections are plotted in orange. The program's numerical result is shown as blue dots. Bottom: The relative difference $\Delta\sigma[\%]$ is shown as black dots. Each column shows the DCS for a different energy E.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

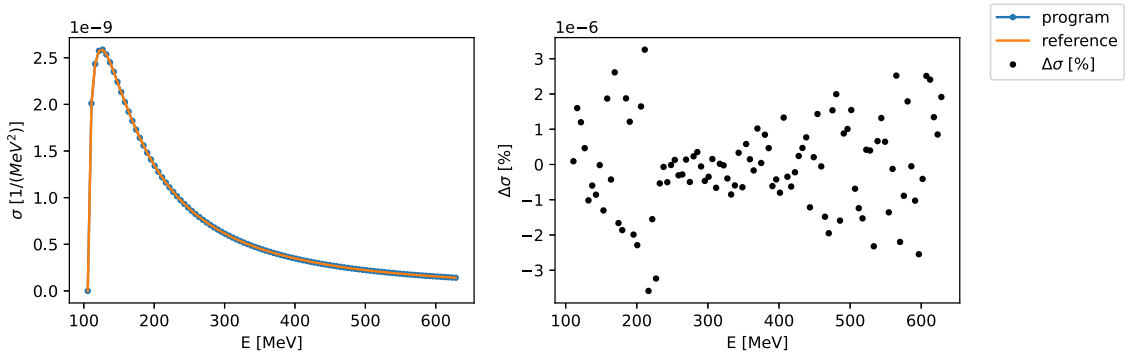


Fig. 5. Left: TCS σ for $e^+e^- \to \mu^+\mu^-$ in the center-of-mass frame in natural units. The analytic cross section is plotted in orange. The program's numerical result is shown as blue dots. Right: The relative difference $\Delta\sigma[\%]$.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

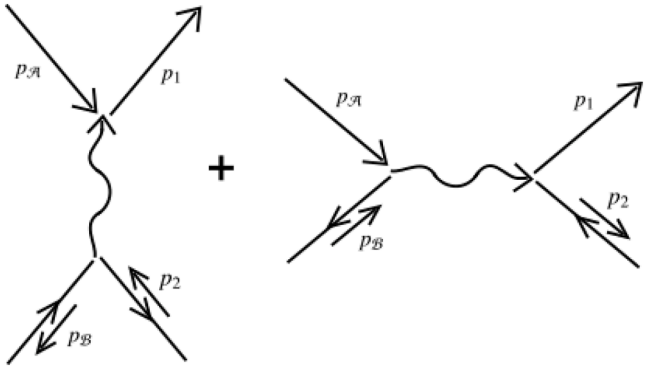


Fig. 6. First order Feynman diagrams for Bhabha scattering [44]. Here $\mathcal{A}$ is the initial electron, $\mathcal{B}$ is the initial positron, particle 1 is the final electron and particle 2 is the final positron. There is one fermion exchange between the diagrams, between p_2 and $p_{\mathcal{A}}$, turning the plus into a minus for the second diagram, when expressing $\mathcal{M}$.

The process is calculated in the center-of-mass frame. We define θ to be the angle between the electron and either one of the final photons, with the electron moving along the z-axis. The analytical solution for the DCS is [28]

$$\frac{d\sigma}{d\Omega} = \frac{\alpha^2}{s}\frac{E}{|\mathbf{p}_{\mathcal{A}}|}\left(\frac{E^2+|\mathbf{p}_{\mathcal{A}}|^2\cos^2\theta}{m_e^2+|\mathbf{p}_{\mathcal{A}}|^2\sin^2\theta} + \frac{2m_e^2}{m_e^2+|\mathbf{p}_{\mathcal{A}}|^2\sin^2\theta} - \frac{2m_e^4}{(m_e^2+|\mathbf{p}_{\mathcal{A}}|^2\sin^2\theta)^2}\right) \tag{19}$$

with the Mandelstam variable $s = (p_{\mathcal{A}} + p_{\mathcal{B}})^2$. Results are shown in Figs. 10 and 11 for various energies. As before, there is an excellent agreement between numerical and analytic results, within a few subpercent. The energy E can be defined here as being the energy of any of the particles since this collision process is considered in the center-of-mass frame.

3.4. Compton scattering: $\gamma\, e^- \to \gamma\, e^-$

The first order Feynman diagram for this process consists of the two diagrams shown in Fig. 12. We use this process to test the program's handling of laboratory frame calculations and SI unit conversion.

We define θ to be the angle between the initial and final photon, with the initial one moving along the z-axis. The initial electron is the stationary particle. The exact reference expression for the Compton scattering DCS is the Klein-Nishina equation [45,46]

$$\frac{d\sigma}{d\Omega} = \frac{1}{2}r_e^2\left(\frac{\omega'}{\omega}\right)^2\left[\frac{\omega'}{\omega} + \frac{\omega}{\omega'} - \sin(\theta)^2\right] \tag{20}$$

where $r_e \approx 2.82 \cdot 10^{-15}$ m is classical electron radius, ω the photon frequency in its initial state and ω' the photon frequency in its final state. The photon frequencies before and after scattering are related through

$$\omega' = \frac{\omega}{1 + \frac{\hbar\omega}{m_e c^2}(1-\cos\theta)} \tag{21}$$

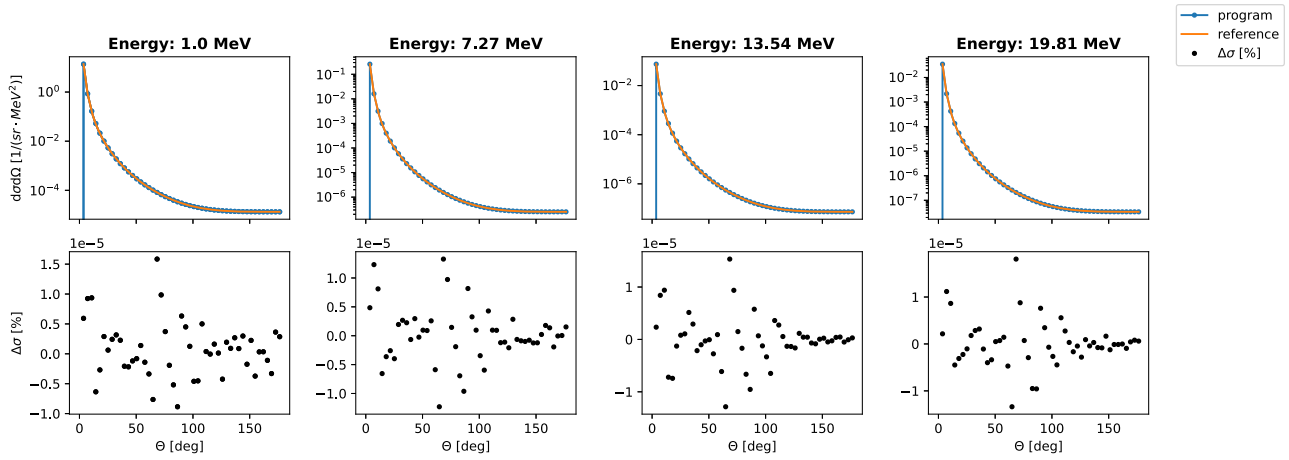


Fig. 7. Top: DCS $d\sigma/d\Omega$ (Eq. (17)) for Bhabha scattering in the center-of-mass frame in natural units. The analytic cross sections are plotted in orange. The program's numerical result is shown as blue dots. Bottom: The relative difference $\Delta\sigma[\%]$ is shown as black dots. Each column shows the DCS for a different energy E.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

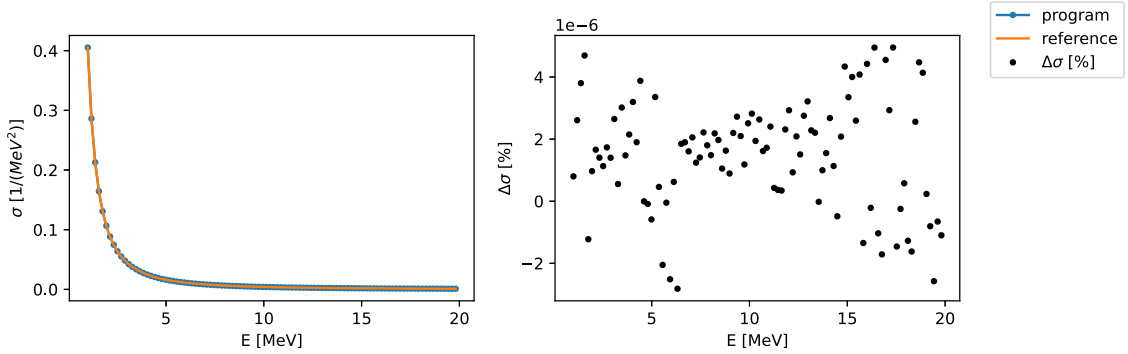


Fig. 8. Left: TCS σ for Bhabha scattering in the center-of-mass frame in natural units. The analytic cross section is plotted in orange. The program's numerical result is shown as blue dots. Right: The relative difference $\Delta\sigma[\%]$.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

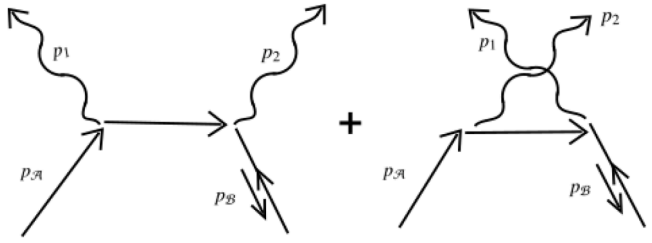


Fig. 9. First order Feynman diagrams for pair annihilation. Here $\mathcal{A}$ is the electron, $\mathcal{B}$ is the positron, and particles 1 and 2 are the resulting photons. The photons are considered indistinguishable final particles.

where $m_e \approx 9.1 \cdot 10^{-31}$ kg is the electron's rest mass. Results in Figs. 13 and 14 show a very good agreement between the numerical and analytical values for various energies. The energy E is defined as the initial photon's energy.

3.5. Comparative discussion

In all four test processes, we see an excellent agreement between the reference and the numerical results, in both DCS and TCS. The relative difference is in the order of $10^{-4}\% - 10^{-6}\%$ for all, with no clearly perceivable structure to it. These small and seemingly random differences suggest them to be of numerically insignificant nature. This suggests a good functioning of our code. Since the differences are small, we observe a high accuracy of the datasets.

We further remark on the symmetry of the pair annihilation DCS. The green symmetry line in Fig. 10 shows the DCS to be symmetric in the ranges $0 \leq \theta \leq \pi/2$ and $\pi/2 \leq \theta \leq \pi$, as expected for a process with two identical final particles.

Finally, let us discuss the energy dependence of the processes. Physically, a process should not be possible if the energy specified in the code for a collision process is less than the total initial rest energy of all considered particles. This is well reflected in the numerically calculated cross sections as both the DCS and TCS go to zero below the initial rest energy. In most cases this means $E \geq m_e = 0.511$ MeV, the rest energy of an electron (in natural units). In the case of $e^+e^- \to \mu^-\mu^+$, $E \geq m_\mu = 105.6$ MeV, the muon's rest mass.

4. Photodissociation

As a first application to our numerical tool, we are presenting first-order results for the photodissociation of molecular nitrogen for which there is only sparse data [33]. This process splits molecular nitrogen into two nitrogen atoms and is fundamentally different from photonuclear processes where a neutron or proton is liberated from a nucleus [26].

The photodissociation of molecules is a slightly different type of process than the previously shown processes since each molecule itself consists of multiple smaller particles, with their own interactions and influences on the cross sections. However, even though a full calculation of all these interactions is currently beyond the scope of our program,

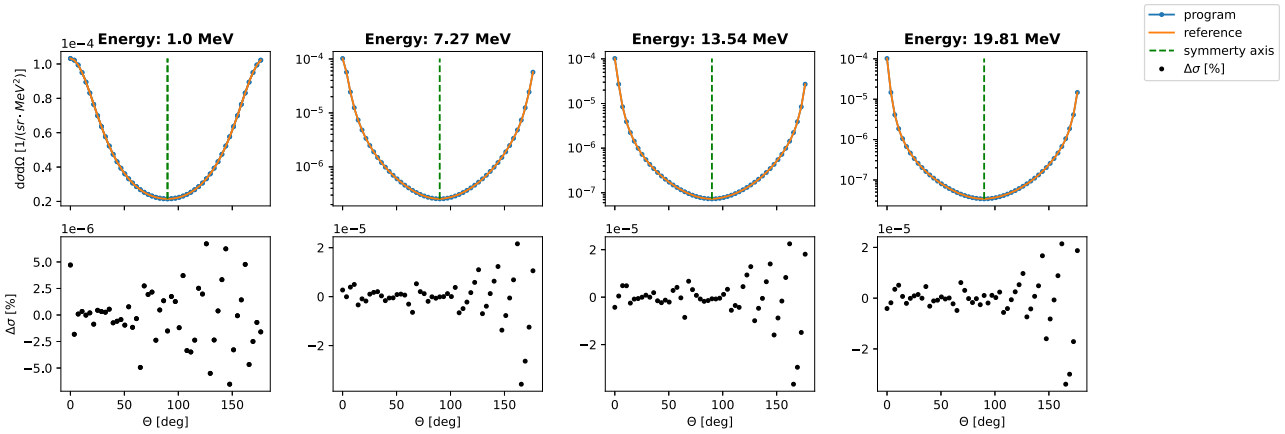


Fig. 10. Top: DCS $d\sigma/d\Omega$ (Eq. (19)) for pair annihilation in the center-of-mass frame in natural units. The analytic cross sections are plotted in orange. The program's numerical result is shown as blue dots. Bottom: The relative difference $\Delta\sigma[\%]$ is shown as black dots. Each column shows the DCS for a different energy E. The dashed green line indicates the symmetry axis at $\theta = \pi/2$ in the DCS, due to identical final particles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

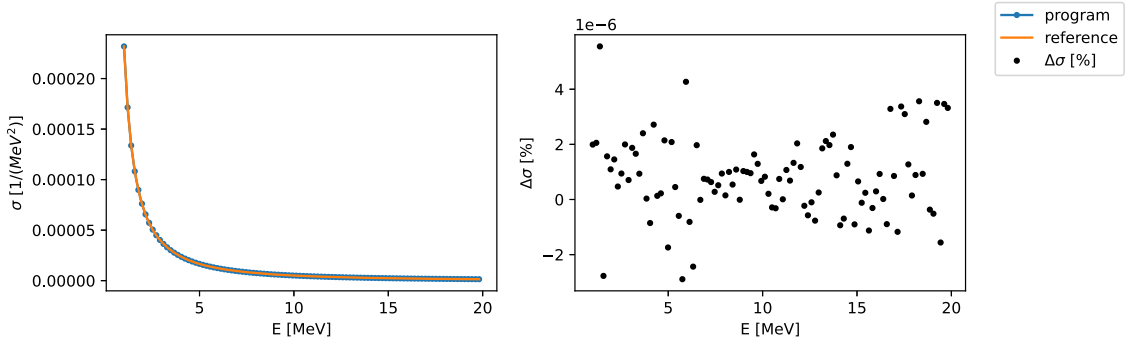


Fig. 11. Left: TCS σ for pair annihilation in the center-of-mass frame in natural units. The analytic cross section is plotted in orange. The program's numerical result is shown as blue dots. Right: The relative difference $\Delta\sigma[\%]$.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

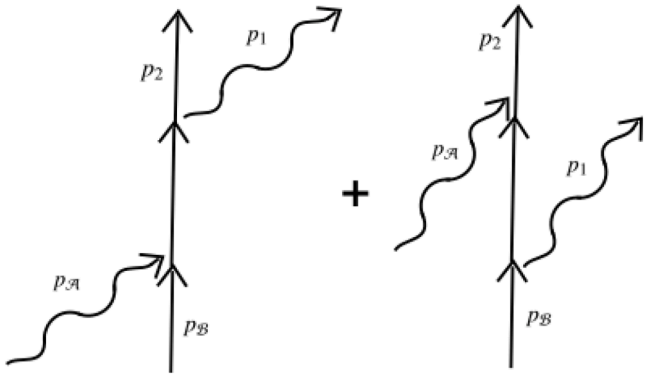


Fig. 12. First order Feynman diagrams for Compton scattering. Here $\mathcal{A}$ is the initial photon, $\mathcal{B}$ is the initial electron, particle 1 is the final photon, and particle 2 is the final electron.

the code is still suited to create accurate first-order cross section data when we consider molecular nitrogen N_2 and atomic nitrogen N as single particle fermions. This is similar to other QED processes of compound particles, such as the scattering of charged pions or the Compton scattering of photons at charged pions [47]. As an example, we will consider the process γ+$N_2 \to 2N$ with the first-order Feynman diagram shown in Fig. 15a.

In order to numerically calculate the cross sections, we need an expression for the 4-momenta of this process. The situation is sketched in Fig. 15b, where we defined the incoming photon to have a momentum $p_\mathcal{A} = (E_\mathcal{A}, 0, 0, E_\mathcal{A})$ and the initial molecule to have a momentum $p_\mathcal{B} = (m_\mathcal{B}, 0, 0, 0)$, i.e. being at rest.

We use energy and momentum conservation to find expressions for the final momenta. The combination of the two conservations results in

$$\begin{aligned} E_\mathcal{A} - E_b + m_\mathcal{B} &= E_1 + E_2 \\ &= \sqrt{|\mathbf{p}_1|^2 + m_1^2} \\ &\quad + \sqrt{|\mathbf{p}_\mathcal{A}|^2 + |\mathbf{p}_1|^2 - 2|\mathbf{p}_\mathcal{A}||\mathbf{p}_1|\cos(\theta) + m_2^2} \end{aligned} \tag{22}$$

for the situation as sketched in Fig. 15b. E_b is the binding energy of the initial molecule $\mathcal{B}$. Eq. (22) can be rewritten as

$$\begin{aligned} &0 = (4|\mathbf{p}_\mathcal{A}|^2\cos(\theta)^2 + 4E_I^2)|\mathbf{p}_1|^2 - 4\square|\mathbf{p}_\mathcal{A}|\cos(\theta)|\mathbf{p}_1| + \square^2 - 4E_I^2 m_1^2, \\ &\text{where } E_I = E_\mathcal{A} - E_b + m_\mathcal{B}, \\ &\text{and } \square^2 = E_I^2 + m_1^2 - |\mathbf{p}_\mathcal{A}|^2 - m_2^2. \end{aligned} \tag{23}$$

This is a standard quadratic equation in $|\mathbf{p}_1|$; we leave it in this general form such that it can be applied to other $2 \to 2$ photodissociation processes as well. The solution is calculated in the input of the program in the functions for the momenta. We only consider the positive solution since $|\mathbf{p}_1|$ must be positive. The 4-momenta then follow to be

$$p_1 = \left(\sqrt{|\mathbf{p}_1|^2 + m_1^2}, |\mathbf{p}_1|\sin(\theta), 0, |\mathbf{p}_1|\cos(\theta)\right) \tag{24}$$

$$p_2 = p_\mathcal{A} + p_\mathcal{B} - p_1 \tag{25}$$

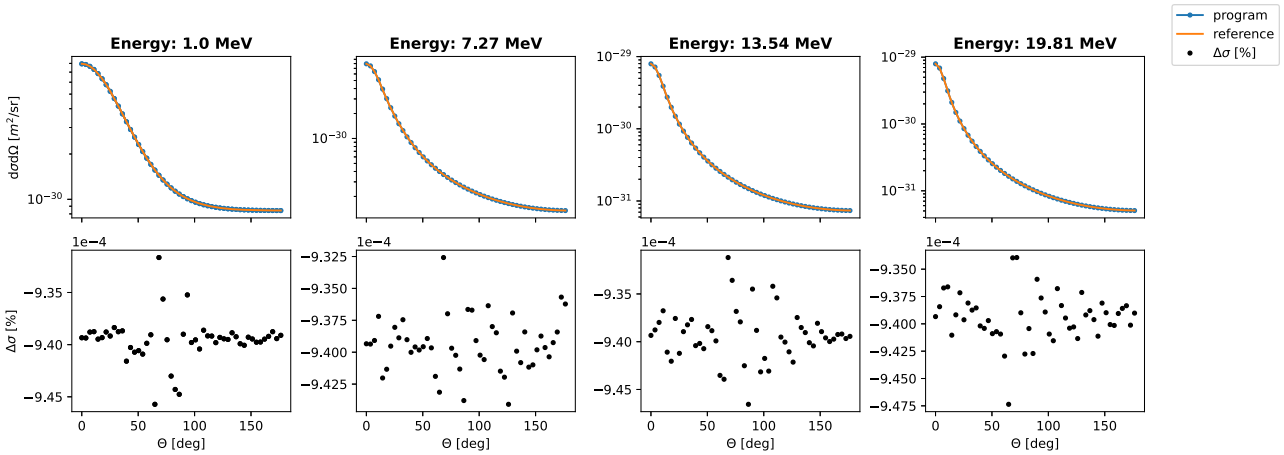


Fig. 13. Top: DCS $d\sigma/d\Omega$ (Eq. (20)) for Compton scattering in the laboratory frame in SI units. The analytic cross sections are plotted in orange. The program's numerical result is shown as blue dots. Bottom: The relative difference $\Delta\sigma[\%]$ is shown as black dots. Each column shows the DCS for a different energy E. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

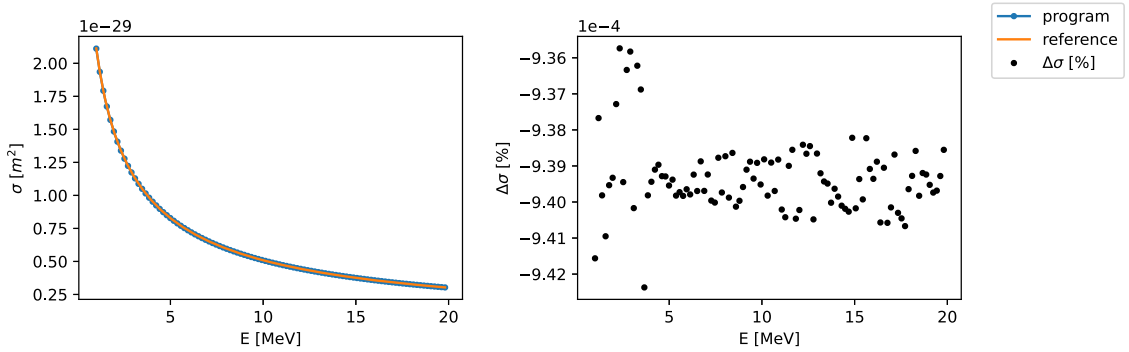
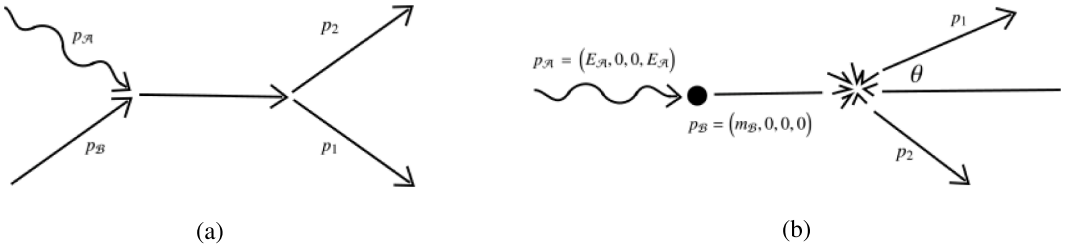


Fig. 14. Left: TCS σ for Compton scattering in the center-of-mass frame in natural units. The analytic cross section is plotted in orange. The program's numerical result is shown as blue dots. Right: The relative difference $\Delta\sigma[\%]$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



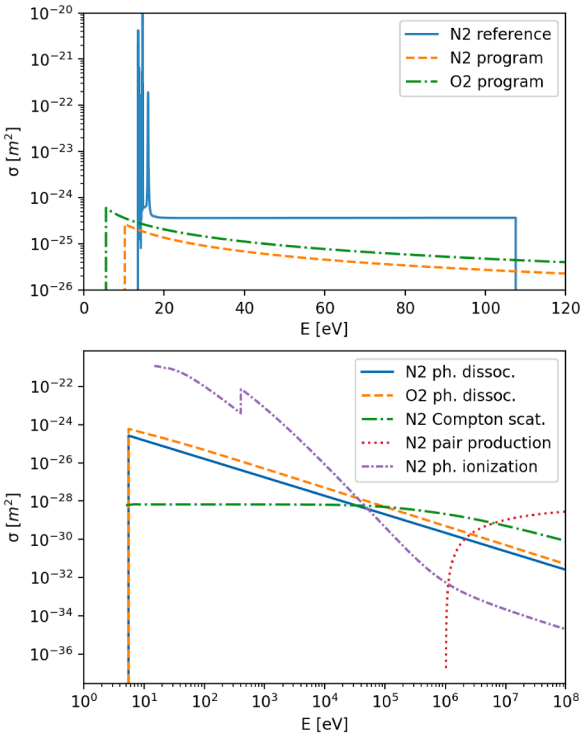
(a) (b)

Fig. 15. (a) First order Feynman diagrams for photodissociation. Here $\mathcal{A}$ is the initial photon, $\mathcal{B}$ is the initial molecule, and particle 1 and 2 are the final molecules. (b) Collision in the laboratory frame for a $2 \to 2$ particle process with one stationary initial particle, and with general expressions for the 4-momenta p. The scattering angle $\theta = \sphericalangle(p_{\mathcal{A}}, p_1)$ is defined as the angle between one of these atoms and the direction of the incident photon.

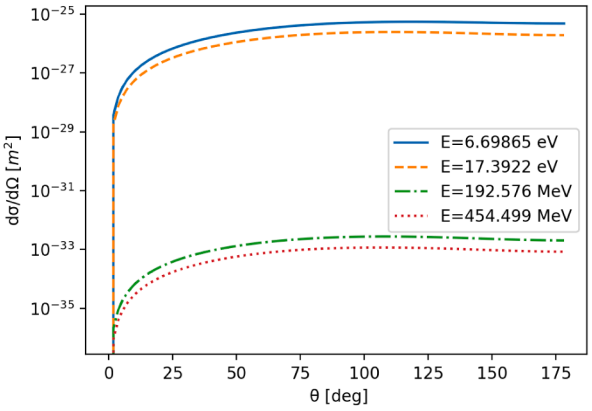
In the case of γ+$N_2 \to 2N$, $m_B = m_{N_2} = 28.0134$ gram/mol $= 26.0943$ MeV (in natural units). Furthermore, $m_1 = m_2 = m_N = m_{N_2}/2$, and for molecular nitrogen $E_b = 9.76 \cdot 10^{-6}$ MeV [48]. Even though the binding energy is included in the momenta, the cross section will not naturally go to zero at energies below this, because we consider N_2 to be one solid particle. To avoid nonphysical results, we add a manual cutoff by having the momentum functions return a zero vector for energies smaller than the binding energy. Finally, since the two final nitrogen molecules are input as the same type of particle with the same mass, they will seem identical to the program. However, they can not be considered as such from a QED point of view, since in reality they consist out of multiple other particles, thus allowing for variations in state and configuration. We therefore run the program without the step of identifying identical particles. The nonidentical nature of the two nitrogen atoms is something we will see confirmed by the lack of angular symmetry in the resulting DCS in Fig. 16b.

The DCS and TCS are shown in Fig. 16, together with those of the photodissociation of O_2 into 2 Os, which was calculated with the exact same input as N_2, but with the appropriate change in mass and binding energy. A reference dataset for the N_2 process' TCS was found in [33] for an energy range of approximately $13.4 - 18$ eV. This data was further extrapolated to the approximate range $13.4 - 108.9$ eV. No reference (literature, dataset) was found for the DCS.

The total cross section (Fig. 16a) by trend agrees with the order of magnitude of the reference cross section. One of the main differences are the resonances for low energies in the reference cross section, whilst the code produces a smooth curve. From the benchmark (Section 3), we know that the cross section calculation of $2 \to 2$ tree-diagram processes



(a)



(b)

Fig. 16. (a) Top: The total photodissociation cross section for N_2 and O_2 as a function of photon energy E, together with the reference for N_2 ([33]). Bottom: The total photodissociation cross section for N_2 and O_2, together with total cross sections for various other photon processes. (b) The differential cross section $d\sigma/d\Omega$ for the photodissociation γ+$N_2 \rightarrow 2N$ for low and high photon energies E.

in itself is very accurate. We therefore suggest that the resonances of the reference cross section originate from higher-order Feynman diagram contributions which are not considered in our code. Although the simplified input results a less detailed TCS, the result is still considered a useful dataset as it allows to calculate the TCS to higher energies in a reasonable manner. Panel a) shows the photodissociation total cross sections for molecular nitrogen and oxygen and compares them to typical cross sections for photoionization, Compton scattering and pair production for molecular nitrogen, thus for the wide energy range from tens of eV up to tens of MeV relevant for beams of relativistic electrons and TGFs. Our calculations show that the cross section for photodissociation is actually on the order of the other processes. Another advantage of our code is that it allows to calculate the differential cross section, see Fig. 16 b), for which no reference data were found, for low and high energies. Furthermore, the angular dependence is not symmetrical in $\pi/2$ suggesting that the 2 N atoms resulting from the process are not necessarily identical particles from a QED point of view, as argued before.

The presented data for photodissociation can be added to typical Monte Carlo simulation codes tracing TGF photons through the atmosphere and usually containing only photoionization, Compton scattering and pair production [20,29,49–51]. Our calculations shows that photodissociation must not be neglected and should be included to give a more realistic picture on how TGFs interact with the atmosphere [30,31].

5. Conclusions and outlook

We have provided a numerical tool that allows to calculate differential and total cross sections for QED processes for a wide range of energies, as they occur in typical TGFs beams; the code is added as Supplementary Material and on `https://gitlab.gbar.dtu.dk/chrstk/cross-section-calculation.git` together with a Wiki on how to use it. From the benchmark results we see that the code is very accurate in calculating cross sections for $2 \rightarrow 2$ tree-diagram processes. We have benchmarked the code with four processes ($e^+e^- \rightarrow \mu^+\mu^-$, Bhabha scattering, pair annihilation, Compton scattering) and observe that both the TCS and DCS agree with the corresponding analytical solutions to within an order of $\sim 10^{-4}$%. These processes involved different combinations of elemental particles, such as photons, electrons, and muons.

As a first application, we have applied the code to the process of photodissociation where we have considered γ+$N_2 \rightarrow 2N$. As a first order approximation, a single tree-level Feynman diagram was used to describe this process, and the molecule and atoms where treated as single "fermions". The resulting TCS showed an overall good agreement with reference data although the code could not resolve resonances which might be attributed to higher-order Feynman diagram contributions. However, we were able to calculate DCS as well as TCS data for higher energies for which there is currently no data. The total cross section is on the order of total cross sections for photoionization, Compton scattering and pair production and should thus be added to particle simulations which are relevant to study the effect of terrestrial gamma-ray flashes or similar high-energy processes onto the atmosphere. With the provided code, the setup of this photodissociation process can be easily copied to processes such as γ+$O_3 \rightarrow O_2$+O, and γ+$NO_2 \rightarrow$NO+ O by changing the input masses and binding energy.

For future development we make the following suggestions:

- Extend the code to calculate multiple final particles, which would include atmospheric processes such as γ+$CO_2 \rightarrow$C+2O, γ+H_2 O$\rightarrow$ 2H + O, and $\gamma + 2NH_3 \rightarrow 3H_2 + N_2$. The main change to achieve this is adapting the phase space integral for the laboratory frame. For two particles, this integral has a fixed analytical expression as presented in Eq. (4). However, for many final particles, the integral

can only be solved using a numerical integration scheme. The code itself requires some minor adjustments in the ordering process, but is generally setup to allow multiple final particles.
- Extend the code to calculate processes with two initially moving particles in the laboratory frame. This would allow for the calculation of processes such as Compton scattering with an moving electron, which is relevant for atmospheric simulations. Currently, the cross sections for this process can be calculated using the center-of-mass frame, after which they can be transformed to other frames. However, as it is now, the code does not allow for the calculation to be done directly in the laboratory frame, since Eq. (4) assumes one stationary particle. As with the previous point, this can be resolved by include a different method of solving for the phase space integral.
- Extend the code to deal with higher order Feynman diagrams. Higher order diagrams might allow for a more detailed result in complex processes such as photodissociation. However, they can take many forms, such as multiple propagators and loops. They would require major extensions to the program, so we leave the details of this option for future work. One example for an extended code could be to calculate cross sections for photonuclear processes, emitting a neutron or proton, and benchmark numerical results against available data [26].

Nevertheless, in cases where there is no reliable data for a given collision process, the provided numerical tool provides suitable cross section approximations that can be implemented into particle Monte Carlo codes for photons and leptons. Additionally, it provides a means to calculate differential and total cross sections where some references only provide total cross sections, and can thus help to model the angular dependence of collision processes correctly. The code allows for the simple calculation of multiple electron and photon processes, and can therefore make more cross sections available for atmospheric simulations than there are at the moment allowing for more accurate simulations of electron and photon beams through the atmosphere. We already know that photonuclear processes have the capability of changing the local atmospheric composition [31], and including additional electron and photon processes for both low and high energies will help us to understand the effect of these species on Earth's atmospheric chemistry.

CRediT authorship contribution statement

H. van Gemert: Writing – original draft, Visualization, Validation, Software, Investigation, Data curation; **F. Defranchi Bisso:** Software, Resources; **C. Köhn:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Funding acquisition, Conceptualization.

Data availability

The data and code is added as supplementary material and also availbale from https://gitlab.gbar.dtu.dk/chrstk/cross-section-calculation.git.

Data and code availability

The program code and its calculated cross section datasets shown in this paper are added as Supplementary Material, or can be downloaded from https://gitlab.gbar.dtu.dk/chrstk/cross-section-calculation.git.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

C.K. and F.D.B. are funded by the Independent Research Fund Denmark (grant 1054-00104). C.K. is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 548340212..

Appendix A. Compiling and executing the code

In order to compile the code, the `eigen` library is needed which are used to perform matrix and vector calculations. In any given location, the eigen library can be downloaded through `git clone https://gitlab.com/libeigen/ eigen.git`. In order to then compile the cross section code, the directory of `eigen` needs to be known.

After downloading the `eigen` library, the following steps need to be performed to compile the cross section code:

- Go into the main folder of the cross section code.
- Run the following commands
  - `mkdir build` (only for the first compiling)
  - `cd build`
  - `cmake . -DCMAKE_PREFIX_PATH =path_to_the_eigen_folder`
  - `cmake --build .`
- Once these steps have been performed, the code can be run through `./cs`.

The Feynman diagrams are encoded in the file `UserInput.h`, see Appendix B below for how to implement Feynman diagrams. Appendix C gives examples for the processes discussed in Sections 3 and 4. Whenever the file `UserInput.h` is changed to implement a different Feynman diagrams and thus calculate different cross sections, the last step (`cmake --build .`) needs to be rerun.

The variables `string pathDCS` and `string pathTCS` specify what the files storing the differential and total cross sections are called. In the file for the differential cross section, the energies E [MeV] are outputted in the first column, the angles θ are outputted in the first row, and the DCS values are tabled as a 2×2 matrix in-between. The total cross section file contains the energies E [MeV] in the first column and the TCS (either in natural units or in SI units, as specified) in the second column.

Appendix B. Encoding Feynman diagrams

Feynman diagrams and some options are encoded in the file `UserInput.h`. A template file with instructions as comments is provided in the folder `input files`, together with the input files for the processes in this paper. In order to use one, copy its content into `UserInput.h`, save, and recompile the code through `cmake --build .`, see Appendix A for details. In order to make a new process, copy the content of the template into `UserInput.h` and replace the examples with the information of the desired process, as described below.

The following can be modified/implemented in the user input file:

- the minimum and maximum energy [MeV]: `double Emin` and `double Emax`
- the energy binning through the divisor in `VectorXd E = arange(Emin, Emax, (Emax - Emin) / 100);` (here 100 as an example). For large energy range, a 10-log binning can be used instead: `VectorXd E = log_arange(10log(Emin), 10log(Emax), 10(EmaxOrder - EminOrder) / 100);`
- number of bins for θ integration: through divisor in `VectorXd theta = arange(0., PI, PI / 100);`, (here 100 as an example)
- the file names of the output files: `string pathTCS` and `string pathDCS`
- choice of center-of-mass frame: `bool CMFrame`; `true` for center-of-mass frame, `false` for laboratory frame
- unit system: `bool natural`; `true` for natural units, `false` for SI units

- momenta functions (paragraph `MOMENTUM VECTORS p`): define `Vector4d` functions returning a four-dimensional momentum vector as a function of E and θ (if spatial momentum components are needed, express them through E and θ); these are used as inputs for the external particles and the propagators. Note that propagators have different momentum expressions in different diagrams
- external particles (paragraph `EXTERNAL PARTICLES`): defining the external particles (user-defined names) of a Feynman diagram including type (antifermion, photon, fermion), direction (entering or exiting vertex), momenta and mass (in natural units).
- propagators (paragraph `PROPAGATORS`): defining propagators (user-defined names) of a Feynman diagram including type (photon, fermion), momenta and mass (natural units); define as many propagators as needed; a propagator needs one `momenta` object that is an array of the momenta of the external particles attached to the first (earlier) vertex that connects to the propagator in the Feynman diagram
- vertices (paragraph `VERTICES`): defining vertices (user-defined names) of a Feynman diagram (define as many vertices as needed); a vertex needs a `particles` object, which is an array of the particles attached to the vertex; also specify which propagator(s) attaches to by using a `propagators` array
- diagrams (paragraph `DIAGRAMS`): defining the `vertices` (based on previous step) and `Diagram` object based on `vertices`; vertices need to be added into a `vertices` array in time order from earliest to latest; define as many diagrams as needed to define the complete process
- define the full collision process (paragraph `PROCESS`) using the diagrams created in the previous step; all diagrams that constitute the full process need to be grouped in the `diagrams` object; the name `process` for the object is fixed

Appendix C. Examples

Example user input files for the Feynman diagrams of the processes discussed in Sections 3 and 4 are provided in the directory `input files`. In order to run these examples, copy the respective `*.txt` file into `UserInput.h`, save, and recompile, see Appendix A. Example results (cross section files and the figures shown in Sections 3 and 4) are provided in the directory `results`.

C.1. $e^+e^- \to \mu^+\mu^-$

The Feynman diagram of this process is shown in Fig. 3a; the momenta for this process are Fig. 3b:

$$p_A = p_{e^-} = (E, 0, 0, E) \tag{C.1}$$
$$p_B = p_{e^+} = (E, 0, 0, -E) \tag{C.2}$$
$$p_1 = p_{\mu^-} = (E, E\sin\theta, 0, E\cos\theta) \tag{C.3}$$
$$p_2 = p_{\mu^+} = (E, -E\sin\theta, 0, -E\cos\theta). \tag{C.4}$$

The input file then looks as here below, where the instruction comments have been omitted for readability. For full instructions and comments on each sections, see the original `.txt` files.

```
#pragma once
#include "classes.h"

////////////7////ELECTRON, POSITRON ->MU+,MU-
    SCATTERING/////////////

/*
GENERAL INPUT PARAMETERS:
* do NOT change general variable names
*
* E      [MeV]   -> energy range; minimal 4 points;
    recommended E > largest_restmass
        linear range: arange(minimum_energy,
            maximum_energy, energy_stepsize)
        log     range   : log_arange(log10(
            minimum_energy), log10(maximum_energy),
            order_stepsize)
* theta[rad]-> angular range; recomemnded 0-PI;
    minimal 4 points;
* pathTCS       -> string name for output total
    cross section file;
* pathDCS       -> string name for output
    differential cross section file;
* CMFrame       -> frame choice;
        true : input and output in center of mass
            frame
        false: input and output in laboratory frame
* natural       -> units choice;
        true : input and output in natural units
        false: input in natural units, output in SI
            units
*/
double Emin = 1. *mmu;
double Emax = 6. *mmu;
VectorXd E = arange(Emin, Emax, (Emax - Emin) / 100)
    ;
VectorXd theta = arange(0., PI, PI / 50);
string pathTCS = "TCS_pair_to_2mumu_CM_natural_units
    .txt";
string pathDCS = "DCS_pair_to_2mumu_CM_natural_units
    .txt";
bool CMFrame = true;
bool natural = true;

/*
MOMENTUM VECTORS p:
* fucntion format: Vector4d func_name(double E,
    double th) {\ldots }
* do not change function parameters
* do not change return type
* calculate momenta in terms of angle theta (th) and
    energy E, matching your general input
    definitions
* inversion of momenta direction between diagrams
    requires a separate function
*/
Vector4d p_A(double E, double th) {
        return Vector4d(E, 0., 0., E);
}
Vector4d p_B(double E, double th) {
        return Vector4d(E, 0., 0., -E);
}
Vector4d p_1(double E, double th) {
        double k3Norm = sqrt(pow(E, 2) - pow(mmu, 2)
            );
        return Vector4d(E, 0., k3Norm * sin(th),
            k3Norm * cos(th));
}
Vector4d p_2(double E, double th) {
        double k3Norm = sqrt(pow(E, 2) - pow(mmu, 2)
            );
        return Vector4d(E, 0., -k3Norm * sin(th), -
            k3Norm * cos(th));
}
```

```
/*
EXTERNAL PARTICLES:
*momenta        -> create array of functions; Total
    particle momentum is consider the sum of those
    functions;
        format: momenta p = {func1, funct2, \ldots
            };
*Particle       -> create particles with their
    properties;
        format: Particle name(arg1, arg2, arg3, arg4
            );
                arg1: type [int]; -1=antifermion, 0=
                    photon, 1=fermion
                arg2: direction [int]; -1 = momentum
                     away from vertex, 0=propagator,
                     1 momentum towards vertex
                arg3: momentum [momenta object];
                    function name with momentum
                    expression
                arg4: mass [double, natural units];
                        Predefined variables: me=
                            electron, mmu=muon, mph=
                            photons, mn=neutron, mp=
                            proton, mtau=tauon
*/
momenta p_El_i = { p_A };
momenta p_Pos_i = { p_B };
momenta p_mu1_f = { p_1 };
momenta p_mu2_f = { p_2 };
Particle El_i(1, 1, p_El_i, me);
Particle Pos_i(-1, 1, p_Pos_i, me);
Particle mu1_f(1, -1, p_mu1_f, me);
Particle mu2_f(-1, -1, p_mu2_f, me);

/*
PROPAGATORS:
* contruct one propagator for each diagram + 1 empty
    propagator
* contruction is done similar to external aprticles,
    see above
* the empty propagator is constructed without
    arguments: Particle emptyProp;
* note: momenta arrays & functions are different for
    different diagrams
*/
momenta p_prop_D1 = { p_A,p_B };
Particle prop_D1(0, 0, p_prop_D1, mph);
Particle emptyProp;

/*
VERTICES:
* For each vertex in each diagram:
*       Group external particles;
                format: particles name = { &
                    Particle1, &Particle1, \ldots };
*       Group propagators LEAVING the vertex (i.e.
    propagators going to a vertex at a later time);
                format propagators name = { &
                    propagator1, &propagator2, \
                    ldots  };
                Use the empty propagator if the
                    vertex is the last in a diagram
                    (i.e. has latest time)
*       Create vertex:
                format: Vertex name(
                    external_particles_group,
                    propagator(s)_group );
*/
particles ex_V1_D1 = { &El_i,&Pos_i };
propagators prop_V1_D1 = { &prop_D1 };
Vertex V1_D1(ex_V1_D1, prop_V1_D1);

particles ex_V2_D1 = { &mu1_f,&mu2_f };
propagators prop_V2_D1 = { &emptyProp };
Vertex V2_D1(ex_V2_D1, prop_V2_D1);

/*
DIAGRAMS:
* For each diagram:
*       Groups vertices TIME ORDERED, from EARLY TO
    LATE;
                format: vertices name = { &vertex1,
                    &vertex2, \ldots  };
*       Create diagram:
*               format: Diagram name(vertices,
    number_of_fermion_exchanges);
*/
vertices vertices_D1 = { &V1_D1,&V2_D1 };
Diagram diagram1(vertices_D1, 0);

/*
PROCESS:
* Group diagram:
        format: diagrams name = { &diag1, &diag2, \
            ldots  };
* Create process:
        format: Process process(diagrams);

DO NOT CHANGE THE VARIABLE NAME "process"
*/
diagrams Diagrams = { &diagram1};
Process process(Diagrams); //do NOT rename "process"
```

C.2. Bhabha scattering: $e^+e^- \to e^+e^-$

The Feynman diagram consists of two diagrams (Fig. 6). The momenta are (in the center-of-mass frame):

$$p_{e^-,initial} = (E, 0, 0, E) \tag{C.5}$$

$$p_{e^+,initial} = (E, 0, 0, -E) \tag{C.6}$$

$$p_{e^-,final} = (E, E\sin\theta, 0, E\cos\theta) \tag{C.7}$$

$$p_{e^+,final} = (E, -E\sin\theta, 0, -E\cos\theta). \tag{C.8}$$

The respective input file then looks like:

```
#pragma once
#include "classes.h"

////////////////BHABHA SCATTERING
    ////////////////////////////////

/*
GENERAL INPUT PARAMETERS:
* do NOT change general variable names
*
* E     [MeV]   -> energy range; minimal 4 points;
    recommended E > largest_restmass
        linear range: arange(minimum_energy,
            maximum_energy, energy_stepsize)
```

```
        log     range   : log_arange(log10(
            minimum_energy), log10(maximum_energy),
            order_stepsize)
* theta[rad]-> angular range; recomemnded 0-PI;
    minimal 4 points;
* pathTCS       -> string name for output total
    cross section file;
* pathDCS       -> string name for output
    differential cross section file;
* CMFrame       -> frame choice;
        true : input and output in center of mass
            frame
        false: input and output in laboratory frame
* natural       -> units choice;
        true : input and output in natural units
        false: input in natural units, output in SI
            units
*/
double Emin = 1.;
double Emax = 20.;
VectorXd E = arange(Emin, Emax, (Emax - Emin) / 100)
    ;
VectorXd theta = arange(0., PI, PI / 50);
string pathTCS = "TCS_bhabha_CM_natural_units.txt";
string pathDCS = "DCS_bhabha_CM_natural_units.txt";
bool CMFrame = true;
bool natural = true;

/*
MOMENTUM VECTORS p:
* fucntion format: Vector4d func_name(double E,
    double th) {\ldots }
* do not change function parameters
* do not change return type
* calculate momenta in terms of angle theta (th) and
     energy E, matching your general input
    definitions
* inversion of momenta direction between diagrams
    requires a separate function
*/
Vector4d p_A(double E, double th) {
        return Vector4d(E, 0., 0., E);
}
Vector4d p_B(double E, double th) {
        return Vector4d(E, 0., 0., -E);
}
Vector4d p_1(double E, double th) {
        return Vector4d(E, E * sin(th), 0., E * cos(
            th));
}
Vector4d p_2(double E, double th) {
        return Vector4d(E, -E * sin(th), 0., -E *
            cos(th));
}
Vector4d inv_p_1(double E, double th) {
        return -1 * Vector4d(E, E * sin(th), 0., E *
             cos(th));
}

/*
EXTERNAL PARTICLES:
*momenta        -> create array of functions; Total
    particle momentum is consider the sum of those
    functions;
        format: momenta p = {func1, funct2, \ldots
            };
*Particle       -> create particles with their
    properties;
        format: Particle name(arg1, arg2, arg3, arg4
            );
                arg1: type [int]; -1=antifermion, 0=
                    photon, 1=fermion
                arg2: direction [int]; -1 = momentum
                     away from vertex,0=propagator,
                    1 momentum towards vertex
                arg3: momentum [momenta object];
                    function name with momentum
                    expression
                arg4: mass [double, natural units];
                        Predefined variables: me=
                            electron, mmu=muon, mph=
                            photons, mn=neutron, mp=
                            proton, mtau=tauon
*/
momenta p_El_i = { p_A };
momenta p_Pos_i = { p_B };
momenta p_El_f = { p_1 };
momenta p_Pos_f = { p_2 };
Particle El_i(1, 1, p_El_i, me);
Particle Pos_i(-1, 1, p_Pos_i, me);
Particle El_f(1, -1, p_El_f, me);
Particle Pos_f(-1, -1, p_Pos_f, me);

/*
PROPAGATORS:
* contruct one propagator for each diagram + 1 empty
     propagator
* contruction is done similar to external aprticles,
     see above
* the empty propagator is constructed without
    arguments: Particle emptyProp;
* note: momenta arrays & functions are different for
     different diagrams
*/
momenta p_prop_D1 = { p_A,p_B };
momenta p_prop_D2 = { p_A, inv_p_1 };
Particle prop_D1(0, 0, p_prop_D1, mph);
Particle prop_D2(0, 0, p_prop_D2, mph);
Particle emptyProp;

/*
VERTICES:
* For each vertex in each diagram:
*       Group external particles;
                format: particles name = { &
                    Particle1, &Particle1, \ldots };
*       Group propagators LEAVING the vertex (i.e.
    propagators going to a vertex at a later time);
                format propagators name = { &
                    propagator1, &propagator2, \
                    ldots  };
                Use the empty propagator if the
                    vertex is the last in a diagram
                    (i.e. has latest time)
*       Create vertex:
                format: Vertex name(
                    external_particles_group,
                    propagator(s)_group );
*/
```

```
particles ex_V1_D1 = { &El_i,&Pos_i };
propagators prop_V1_D1 = { &prop_D1 };
Vertex V1_D1(ex_V1_D1, prop_V1_D1);

particles ex_V2_D1 = { &El_f,&Pos_f };
propagators prop_V2_D1 = { &emptyProp };
Vertex V2_D1(ex_V2_D1, prop_V2_D1);

particles ex_V1_D2 = { &El_i,&El_f };
propagators prop_V1_D2 = { &prop_D2 };
Vertex V1_D2(ex_V1_D2, prop_V1_D2);

particles ex_V2_D2 = { &Pos_i,&Pos_f };
propagators prop_V2_D2 = { &emptyProp };
Vertex V2_D2(ex_V2_D2, prop_V2_D2);

/*
DIAGRAMS:
* For each diagram:
*       Groups vertices TIME ORDERED, from EARLY TO
    LATE;
                format: vertices name = { &vertex1,
                    &vertex2, \ldots  };
*       Create diagram:
*               format: Diagram name(vertices,
    number_of_fermion_exchanges);
*/
vertices vertices_D1 = { &V1_D1,&V2_D1 };
Diagram diagram1(vertices_D1, 0);

vertices vertices_D2 = { &V1_D2,&V2_D2 };
Diagram diagram2(vertices_D2, 1);

/*
PROCESS:
* Group diagram:
        format: diagrams name = { &diag1, &diag2, \
            ldots  };
* Create process:
        format: Process process(diagrams);

DO NOT CHANGE THE VARIABLE NAME "process"
*/
diagrams Diagrams = { &diagram1, &diagram2 };
Process process(Diagrams); //do NOT rename "process"
```

References

[1] G.J. Fishman, et al., Discovery of intense gamma-ray flashes of atmospheric origin, Science 264 (1994) 1313–1316.

[2] M. Marisaldi, F. Fuschino, C. Labanti, M. Galli, F. Longo, E. Del Monte, Detection of terrestrial gamma ray flashes up to 40 MeV by the AGILE satellite, J. Geophys. Res. 115 (2010) A00E13.

[3] M. Tavani, M. Marisaldi, C. Labanti, F. Fuschino, A. Argan, A. Trois, et al., Terrestrial gamma-ray flashes as powerful particle accelerators, Phys. Rev. Lett. 106 (2011) 018501.

[4] M.S. Briggs, S. Xiong, V. Connaughton, D. Tierney, G. Fitzpatrick, S. Foley, et al., Terrestrial gamma-ray flashes in the Fermi era: improved observations and analysis methods, J. Geophys. Res. 118 (2013) 3805–3830.

[5] A. Lindanger, M. Marisaldi, D. Sarria, N. Østgaard, N. Lehtinen, C.A. Skeie, A. Mezentzev, P. Kochkin, K. Ullaland, S. Yang, G. Genov, B.E. Carlson, C. Köhn, J. Navarro-Gonzalez, P. Connell, V. Reglero, T. Neubert, Spectral analysis of individual terrestrial gamma-ray flashes detected by ASIM, J. Geophys. Res. Atmos. 126 (2021) e2021JD035347.

[6] M. Heumesser, et al., Spectral observations of optical emissions associated with terrestrial gamma-ray flashes, Geophys. Res. Lett. 48 (2021) e2020GL090700.

[7] H.A. Bethe, W. Heitler, On the stopping of fast particles and on the creation of positive electrons, Proc. R. Soc. Lond. A 146 (1934) 83–112. https://doi.org/10.1098/rspa.1934.0140

[8] C. Köhn, U. Ebert, Angular distribution of Bremsstrahlung photons and of positrons for calculations of terrestrial gamma-ray flashes and positron beams, Atmos. Res. 135–136 (2014) 432–465. https://doi.org/10.1016/j.atmosres.2013.03.012

[9] J.R. Dwyer, D.M. Smith, S.A. Cummer, High-energy atmospheric physics: terrestrial gamma-ray flashes and related phenomena, Space. Sci. Rev. 173 (2012) 133–196. https://doi.org/10.1007/s11214-012-9894-0

[10] C. Köhn, M. Heumesser, O. Chanrion, K. Nishikawa, V. Reglero, T. Neubert, The emission of terrestrial gamma ray flashes from encountering streamer coronae associated to the breakdown of lightning leaders, Geophys. Res. Lett. 47 (2020) e2020GL089749. https://doi.org/10.1029/2020GL089749

[11] C. Köhn, et al., Recent results on science and innovation related to electrical processes of thunderstorms, Surv. Geophys. 46 (2025) 753–821.

[12] N. Østgaard, et al., Airborne lightning observatory for FEGS and TGFs (ALOFT) (2023). https://www.uib.no/en/aloft.

[13] M.S. Briggs, G.J. Fishman, V. Connaughton, P.N. Bhat, W.S. Paciesas, R.D. Preece, et al., First results on terrestrial gamma ray flashes from the Fermi gamma-ray burst monitor, J. Geophys. Res. 115 (2010) A07323.

[14] T.G. Østgaard N., K. H. Albrecthsen, A. Collier, A new population of terrestrial gamma-ray flashes in the RHESSI data, Geophys. Res. Lett. 42 (2015) 10937–10942.

[15] T. Neubert, et al., The ASIM mission on the international space station, Space Sci. Rev. 215 (2019) 26.

[16] N. Østgaard, et al., First 10 months of TGF observations by ASIM, J. Geophys. Res. Atmos. 124 (2019) 14024–14036.

[17] C. Maiorana, M. Marisaldi, A. Lindanger, N. Østgaard, A. Ursi, D. Sarria, M. Galli, C. Labanti, M. Tavani, C. Pittori, F. Verrecchia, The 3rd AGILE terrestrial gamma-ray flashes catalog. part II: optimized selection criteria and characteristics of the new sample, J. Geophys. Res. Atmos. 125 (2020) e2019JD031986.

[18] V. Vahedi, M. Surendra, A Monte Carlo collision model for the particle-in-cell method applications to argon and oxygen discharges, Comp. Phys. Comm. 87 (1995) 179–198.

[19] O. Chanrion, T. Neubert, A PIC-MCC code for simulations of streamer propagation in air, J. Comp. Phys. 227 (2008) 7222–7245.

[20] C. Köhn, U. Ebert, Calculation of beams of positrons, neutrons and protons associated with terrestrial gamma-ray flashes, J. Geophys. Res. 120 (2015) 1620–1635.

[21] T. Murphy, Total and differential electron collision cross sections for O_2 and N_2, Report LA-11288-MS Los Alamos National Laboratory (1988).

[22] A. Schmalzried, A. Luque, Influence of elastic scattering on electron swarm distribution in electrified gases, J. Geophys. Res. Atmos. 125 (2020) e2019JD031564.

[23] Y.K. Kim, J.P. Santos, F. Parente, Extension of the binary-encounter-dipole model to relativistic incident electrons, Phys. Rev. A 62 (2000) 052710.

[24] S. Celestin, V.P. Pasko, Soft collisions in relativistic runaway electron avalanches, J. Phys. D Appl. Phys. 43 (2010) 315206.

[25] C. Köhn, U. Ebert, The structure of ionization showers in air generated by electrons with 1 MeV energy or less, Plasma Sources Sci. Technol. 23 (2014) 045001.

[26] E.G. Fuller, Photonuclear reaction cross sections for ^{12}C, ^{14}N and ^{16}O, Phys. Rep. 127 (1985) 185–231.

[27] EPDL, Photon and electron interaction data, 1997. http://www-nds.iaea.org/epdl97.

[28] M.E. Peskin, D.V. Schroeder, An Introduction to Quantum Field Theory, Westview Press, 1995. Reading, USA: Addison-Wesley (1995), 842 p.

[29] C. Köhn, others., Leptons, hadrons and photons and their feedback close to lightning leaders, J. Geophys. Res. 122 (2017a) 1365–1383.

[30] E.S. Cramer, M.S. Briggs, N. Liu, B. Mailyan, J.R. Dwyer, H.K. Rassoul, The impact on the ozone layer fromNOx produced by terrestrial gamma ray flashes, Geophys. Res. Lett. 44 (2017) 5240–5245.

[31] T. Enoto, Y. Wada, Y. Furuta, K. Nakazawa, T. Yuasa, K. Okuda, K. Makishima, M. Sato, Y. Sato, T. Nakano, D. Umemoto, H. Tsuchiya, Photonuclear reactions triggered by lightning discharge, Nature 44 (2017) 551.

[32] J.H. Waite, Jr., D.T. Young, T.E. Cravens, A.J. Coates, F.J. Crary, B. Magee, J. Westlake, The process of Tholin formation in Titan's upper atmosphere, Science 316 (2007) 870–875.

[33] A. Li, A.N. Heays, R. Visser, W. Ubachs, B.R. Lewis, E.F. S. T. van Dishoeck, Gibson, Photodissociation of interstellar N_2, Astro. & Astro. 555 (2013) A14.

[34] LXCat, LXCAT: Phelps database, 2026, http://www.lxcat.net.

[35] C.B. Opal, W.K. Peterson, E.C. Beaty, Measurements of secondary-electron spectra produced by electron impact ionization of a number of simple gases, J. Chem. Phys. 55 (8) (1971) 4100–4106. https://doi.org/10.1063/1.1676707

[36] G.D. Moss, V.P. Pasko, N. Liu, G. Veronis, Monte Carlo model for analysis of thermal runaway electrons in streamer tips in transient luminous events and streamer zones of lightning leaders, J. Geophys. Res. Space Phys. 111 (A2) (2006). https://doi.org/10.1029/2005JA011350

[37] R. Harlander, M. Steinhauser, Automatic computation of feynman diagrams, Prog. Part. Nucl. Phys. 43 (1999) 167–228.

[38] T. Hahn, Generating Feynman diagrams and amplitudes with FeynArts 3, Comput. Phys. Commun. 140 (3) (2001) 418–431. https://doi.org/10.1016/S0010-4655(01)00290-9

[39] A.S. Belyaev, E.E. Boos, A.N. Vologdin, M.N. Dubinin, V.A. Ilyin, A.P. Kryukov, A.E. Pukhov, A.N. Skachkova, V.I. Savrin, A.V. Sherstnev, S.A. Shichanin, CompHEP-PYTHIA interface: integrated package for the collision events generation based on exact matrix elements, AIP Conf. Proc. 583 (2001) 211–213. https://doi.org/10.1063/1.1405306

[40] R. Mertig, M. Böhm, A. Denner, Feyn Calc - Computer-algebraic calculation of Feynman amplitudes, Comput. Phys. Commun. 64 (1991) 345–359. https://doi.org/10.1016/0010-4655(91)90130-D

[41] R.P. Feynman, The Strange Theory of Light and Matter, Princeton University Press, Princeton, New Jersey, 1985.
[42] H. Yukawa, On the interaction of elementary particles, Proc. Phys.-Math. Soc. Jpn. 17 (1935) 48.
[43] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: fundamental algorithms for scientific computing in python, Nat. Methods 17 (2020) 261–272. https://doi.org/10.1038/s41592-019-0686-2
[44] A. Santos, F. Khanna, Bhabha scattering in very special relativity at finite temperature, The European Physical Journal C 80 (2020). https://doi.org/10.1140/epjc/s10052-020-8290-2
[45] O. Klein, Y. Nishina, ÜBer die streuung von strahlung durch freie elektronen nach der neuen relativistischen quantendynamik von dirac, Zeitschrift für Physik 52 (11–12) (1929) 853–868. https://doi.org/10.1007/bf01366453
[46] S. Weinberg, 8.7 Compton Scattering, vol. I, Cambridge University Press, 1995, p. 362–369.
[47] W. Greiner, J. Reinhardt, Quantum Electrodynamics, Springer, Berlin, Heidelberg, 2009.
[48] B. Ruscic, D. Feller, K.A. Peterson, Active thermochemical tables: dissociation energies of several homonuclear first-row diatomics and related thermochemical values, Theor. Chem. Acc. 133 (1) (2013) 1415. https://doi.org/10.1007/s00214-013-1415-z.
[49] N. Østgaard, T. Gjesteland, J. Stadsnes, P.H. Connell, B. Carlson, Production altitude and time delays of the terrestrial gamma flashes: revisiting the burst and transient source experiment spectra, J. Geophys. Res. 113 (2008) A02307.
[50] S. Celestin, V.P. Pasko, Energy and fluxes of thermal runaway electrons produced by exponential growth of streamers during the stepping of lightning leaders and in transient luminous events, J. Geophys. Res. 116 (2011) A03315.
[51] X.W. S. Celestin, V.P. Pasko, Source altitudes of terrestrial gamma-ray flashes produced by lightning leaders, Geophys. Res. Lett. 39 (2012) L08801.