Andreas Øren Nossum

Bethe-Heitler emission of electrons from high-energy protons in different astrophysical environments

Bachelor's thesis in Physics
Supervisor: Foteini Oikonomou
June 2023

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU

ABSTRACT

In this thesis, we numerically calculate the spectral energy distribution (SED) of electrons (positrons) from the Bethe-Heitler pair production process, based on the formalism presented by Kelner and Aharonian (2009) [1]. This implementation can be used to further expand `agnpy` [2], which already contains similar functionalities for other particle processes. To begin with, we set the stage for the processes in question by presenting the astrophysical environment where they are likely to occur. Then, after a short summary of relevant relativistic particle concepts, we derive the energy threshold for both pair production and photomeson production, respectively, before examining synchrotron radiation. From this, we present the open source code `agnpy`, and study how the latter radiation is implemented. Next, we dissect the functional form of the SED of electrons from pair production (presented in [1]), and discuss the integration limits thoroughly. We create a numerical implementation of this theoretical framework, and compare our results with the ones in [1]. We find that SEDs produced by our implementation are in excellent agreement with those of [1]. Finally, we discuss the efficiency of our code, along with future prospects and the relevance of this process in astrophysical environments.

PREFACE

First and foremost, I would like to express my sincerest appreciation to Foteini Oikonomou. This project would not even be remotely possible without your guidance, and I want to thank you for your patience and encouragement. A special thanks goes to Egor Podlesnyi who has given me sound advice along the way. Finally, I would be remiss in not mentioning my fellow classmates, friends and family. Cheers!

CONTENTS

LIST OF FIGURES

ABBREVIATIONS

List of all abbreviations in alphabetic order:

- **AGN** Active Galactic Nucleus
- **SED** Spectral Energy Distribution
- **CMB** Cosmic Microwave Background
- **COM** Center of Momentum

CHAPTER

ONE

INTRODUCTION

1.1 Motivation

As telescope and detector technology evolves, more and more celestial objects become possible to observe. One such object, which is not yet fully understood, is an active galactic nucleus (AGN). Therefore, being able to examine the signals received from an AGN is desired, since it inversely gives information about what is brewing inside the object itself. For instance, one could determine which particle processes occur based on the observed energy spectrum. However, when the actual data is in short supply, a theoretical model could possibly give sufficient, if not greater, insight into how the signature of each particle process appears on such a spectrum. Thus, one could verify future data based on previously established models. By means of this, computational calculations and modeling serves as some of the many strengths of astrophysics. Still, as many physicists will agree on, without the use of already existing software, computational modeling would pose a greater challenge. Many of these softwares are the results of smaller contributions, and are usually publicly available. Thus, as more and more computational needs are accounted for, the larger the toolbox of astrophysics grows.

1.2 Project description

In this project, we intend to numerically calculate the SED of electrons from the Bethe-Heitler pair production process, based on the formalism presented in Kelner and Aharonian (2009) [1]. In doing so, we will make use of the publicly available source code named `agnpy` [2]. This software is a `python` package used to model radiative processes of jetted AGNs, and contains tutorials on how to calculate SEDs for some processes, initialize particle distributions, and create emission regions, to mention a few. For instance, one could easily utilize this source code to model SEDs from synchrotron radiation or photon-photon absorption. However, not every particle process or radiation is included. On that account, our numerical implementation of pair production, based on [1], could further expand the software.

CHAPTER

TWO

BACKGROUND

2.1 Astrophysical environments

First of all, we require an understanding of the astrophysical environments which `agnpy` is based upon. Accordingly, we will look at a general active galaxy, describe its spectrum, venture into its nucleus, and finally into the jet where one can find the relativistic particle processes discussed in [1].

2.1.1 Active Galactic Nuclei

One way of determining which galaxies are active, is to look at the energy spectra produced by the galaxy as a whole. Such a spectrum shows the measure of energy for each point of the scale, and the active galaxies will have a flatter curve than its normal counterpart. Where the normal galaxies show peaks around optical wavelengths, the active ones exhibit more emission for X-ray and radio wavelengths. This, in turn, indicates that something in the active galaxies gives rise to a considerable amount of energy, which is not found in the normal equivalent.

Some types of active galaxies mentioned in [3] are Seyfert galaxies, quasars, radio galaxies and blazars. The naming, classification, and sub classification, can seem somewhat peculiar, but the common factor is the high amount of energy originating specifically from the nucleus of the galaxy. This has, in turn, given rise to the name active galactic nucleus, which is the difference between a normal galaxy and an active one.

According to [3], the current models of an AGN consist of the highly luminous central engine surrounded by an obscuring torus of gas and dust. The central engine is made up of a supermassive black hole, an accretion disc, and two powerful jets streaming energetic particles perpendicular to the disc. So energetic, in fact, that they reach relativistic speeds. Thus, the SED from an active galaxy displays non-thermal particle contributions, hence, a flatter curve than its normal equivalent. Although the entire composition of an AGN is only seen as a single, point-like nucleus on optical images, the jets can extend far beyond its host galaxy. It is in these highly energetic environments that particles undergo the relativistic processes referenced in [1] and (some) implemented in `agnpy`. But before we delve

any further into the particle processes in question, it would be favorable to summarize some key particle physics concepts to make the coming calculations less tedious.

2.2 Key particle physics concepts

In doing so, some important points from the 2nd chapter of Dermer and Menon (2009) [4] is recited. To begin with, we invoke the power of special relativity before considering particle interactions in general. The concept of cross sections will also be introduced.

2.2.1 Special relativity

As in [4], consider two inertial frames K and K' aligned in the $\hat{x}$ direction. The second frame, K', is moving with the speed $v = \beta c$ along $\hat{x}$ relative to the first frame. The postulates of special relativity state that the laws of physics are the same in inertial reference frames, and that light always moves at the speed c. Consequently, the combination of the two postulates gives that an interval satisfies

$$-c^2t^2 + x^2 + y^2 + z^2 = -c^2t'^2 + x'^2 + y'^2 + z'^2 = 0. \tag{2.1}$$

The expression above is denoted Eq. (2.1) in [4] as well, and the primed quantities refer to the moving inertial frame K'. According to an observer in the first frame, it is obvious that the second, primed frame, is the one hurrying away. However, stepping into this proposed "moving" frame, and keeping the first postulate in mind, one can undoubtedly announce that the first frame is the one moving! This stepping back and forth can be expressed by the Lorentz transformations, which is, mathematically speaking, a set of linear equations satisfying Eq. (2.1). Defining the bulk Lorentz factor as $\Gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$, the Lorentz equations – connecting an event $\vec{x}$ and time t in the first frame with the moving one – can be written as such

$$\begin{aligned} t' &= \Gamma(t - \beta x/c) \\ x' &= \Gamma(x - \beta ct) \\ y' &= y \\ z' &= z. \end{aligned} \tag{2.2}$$

One would assume that describing the same relation from the K' frame's point of view would be to rearrange the terms above, but this is where special relativity becomes weird. The inverse transformation is deduced by stepping into the K' frame, and, as mentioned, realizing that the first frame is the one moving, but now in the $-\hat{x}$ direction. It is all about the point of view – the *relative* part of relativity. With this new mindset, and using the sign-change from v to $-v$, the inverse equations are then written as

$$\begin{aligned} t &= \Gamma\left(t' + \beta x'/c\right) \\ x &= \Gamma\left(x' + \beta ct'\right) \\ y &= y' \\ z &= z'. \end{aligned} \tag{2.3}$$

Keep in mind that the velocity has only changed sign, and the factor Γ is therefore unchanged due to the v^2 dependence. The transformation, and inverse transformation, recited here, are only along one direction. This is generally referred to as a "boost". Special relativity is most useful when applied to particle interactions.

2.2.2 Relativistic kinematics

A particle in space-time can be described with a four-vector $\vec{X} = (ct, \vec{x}) = (x^0, x^1, x^2, x^3)$. And as stated by the Lorentz transformation above, in a moving frame, this vector transforms as such

$$\begin{aligned} x'^0 &= \gamma\left(x^0 - \beta x^1\right) \\ x'^1 &= \gamma\left(x^1 - \beta x^0\right) \\ x'^2 &= x^2 \\ x'^3 &= x^3, \end{aligned} \tag{2.4}$$

where the the particle moves with the speed v, and therefore $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. Consequently, the four-momentum of the particle is then

$$\vec{P} = \gamma m \frac{\partial \vec{X}}{\partial t} = \gamma m \frac{\partial}{\partial t}(ct, \vec{x}) = (\gamma mc, \gamma m\vec{v}). \tag{2.5}$$

Using the fact that $E = mc^2 + (\gamma - 1)mc^2 = \gamma mc^2$ (a contribution of rest- and kinetic energy) and $\vec{p} = \gamma m\vec{v}$, Eq. (2.5) becomes

$$\vec{P} = \left(\frac{E}{c}, \vec{p}\right) = \left(\frac{E}{c}, p_x, p_y, p_z\right). \tag{2.6}$$

This is also a four-vector, and transforms as such

$$\begin{aligned} \frac{E'}{c} &= \gamma\left(\frac{E}{c} - \beta p_x\right) \\ p'_x &= \gamma\left(p_x - \beta\frac{E}{c}\right) \\ p'_y &= p_y \\ p'^z_z &= p_z. \end{aligned} \tag{2.7}$$

The inverse transformation can be, not surprisingly, written as

$$\begin{aligned} \frac{E}{c} &= \gamma\left(\frac{E'}{c} + \beta p'_x\right) \\ p_x &= \gamma\left(p'_x + \beta\frac{E'}{c}\right) \\ p_y &= p'_y \\ p_z &= p'_z. \end{aligned} \tag{2.8}$$

Finally, it can also be shown, with the usual convention of vector products in special relativity, that the squared four-momentum equals

$$\vec{P} \cdot \vec{P} = -\frac{E^2}{c^2} + \|\vec{p}\|^2 = -m^2c^2, \tag{2.9}$$

often called the energy-momentum relation. Where the classical conservation laws state that mass and momentum are conserved, the relativistic equivalent is saying that four-momentum is conserved. This conservation law, along with Eq. (2.9), will be especially important when considering the kinematics and energy thresholds for some specific particle interactions.

2.2.3 Cross section

The cross section gives, in a crude sense, insight into how probable the specific reaction is (see e.g. [5]). It is usually denoted σ and with units of area. The cross section is also dependent on the scattering angle of the colliding components, thus, uniquely defined for each particle process. Therefore, the individual cross sections are crucial when dealing with specific particle collisions. With this brief introduction of relativistic particle physics, the stage is set for the specific particle processes which are found in jetted AGNs.

2.3 Relevant particle processes

In the high energy regime, one of the main processes of light-matter interaction is that of pair production. This particle production process will be the main concern of this thesis. Other processes which occur for lower energies include, among others, Compton scattering, Thompson scattering and photoelectric effect. However, when studying ultra relativistic protons in astrophysical environments where the density of lower energy radiation surpasses the density of the gas component, the main reactions with radiation happens through inverse Compton scattering, pair production and photomeson production (see e.g. [1]). To begin with, we examine the photomeson process before venturing into pair production and synchrotron radiation.

2.3.1 Photomeson production

The photomeson process is a photohadronic process and denotes the creation of a meson, often a pion, from the photon-nucleus interaction $N + \gamma \rightarrow N + \pi$. The production of a charged pion is equally probable as a neutral one, but bear in mind that for conservation of charge to be valid, a neutral or charged pion has to be accompanied by either a proton or neutron, respectively. What makes this production spark particular interest, is the fact that the reaction can further produce neutrinos – an important piece in identifying cosmic ray accelerators (see e.g. [4]). From a proton-photon interaction, both neutrinos, denoted ν, and γ-rays can be produced through the channels

$$p + \gamma \rightarrow \begin{cases} p + \pi^0 \rightarrow p + 2\gamma \\ n + \pi^+ \rightarrow n + \mu^+ + \nu_+ \\ n + \pi^- \rightarrow n + \mu^- + \bar{\nu}_-, \end{cases} \tag{2.10}$$

as shown by Eq. (8.53) and Eq. (8.54) in [4]. Also shown, is that the last two channels with charged pions can further decay into

$$\begin{cases} n + \mu^+ + \nu_+ \rightarrow n + e^+ + \bar{\nu}_\mu + \nu_e + \nu_+ \\ n + \mu^- + \bar{\nu}_- \rightarrow n + e^- + \nu_\mu + \bar{\nu}_\mu + \bar{\nu}_-, \end{cases} \tag{2.11}$$

hence, creating a multitude of different neutrinos. Along the neutral pion channel, however, two γ-rays are created. These can, in turn, create an electron-positron pair according to pair production. An initially simple reaction can, under the right condition, cascade into a zoological garden of different neutrino flavors and particle productions. It is therefore important to examine which initial photon energy leads to which reaction.

Photomeson production is a threshold reaction, meaning that a certain energy is required for the reaction to take place. One can determine this respective threshold energy by looking at the kinematic relation between the proton and photon energy in the production of a neutral pion. We will do so by considering a head-on collision between the proton and photon in the laboratory frame, and start by denoting each particle's four-momentum as

$$\vec{P}_p = \left(\frac{E_p}{c}, -p_p, 0, 0 \right) \tag{2.12}$$

$$\vec{P}_\gamma = \left(\frac{E_\gamma}{c}, \frac{E_\gamma}{c}, 0, 0 \right). \tag{2.13}$$

Note here that the momenta point towards each other, and the momentum of a photon is given by $p_\gamma = \frac{E_\gamma}{c}$. Hence, the square of total four-momentum before the collision is given as

$$(\vec{P}_{lab}^{sys})^2 = \vec{P}_p^2 + 2\vec{P}_p\vec{P}_\gamma + \vec{P}_\gamma^2 = (-m_p^2c^2) + 2(-\frac{E_pE_\gamma}{c^2} - \frac{E_\gamma}{c}p_p) + (0). \tag{2.14}$$

An important observation is that the square of a photon's four-momentum falls away in the given convention. A tidier version of Eq. (2.14) is written as

$$(\vec{P}_{lab}^{sys})^2 = -m_p^2c^2 - 2\frac{E_pE_\gamma}{c^2} - 2\frac{E_\gamma}{c}p_p. \tag{2.15}$$

The same treatment will be given to the system after the collision in a center-of-momentum frame (COM frame). The four-momentum of the resulting proton and pion in this frame can be written as

$$\vec{P}_{p'} = \left(\frac{E_{p'}}{c}, \vec{p}_{p'} \right) \tag{2.16}$$

$$\vec{P}_{\pi^0} = \left(\frac{E_{\pi^0}}{c}, \vec{p}_{\pi^0} \right) \tag{2.17}$$

Note here that a COM frame is by definition a frame where the sum of $\vec{p}_{p'}$ and $\vec{p}_{\pi^0}$ equals the zero vector, such that

$$\vec{P}^{sys}_{COM} = \vec{P}_{p'} + \vec{P}_{\pi^0} = \left(\frac{E_{p'}}{c} + \frac{E_{\pi^0}}{c}, \vec{p}_{p'} + \vec{p}_{\pi^0}\right) = \left(\frac{E_{p'}}{c} + \frac{E_{\pi^0}}{c}, 0\right). \tag{2.18}$$

Thus, squaring this equation as before simply yields

$$(\vec{P}^{sys}_{COM})^2 = \left(\frac{E_{p'}}{c} + \frac{E_{\pi^0}}{c}\right)^2 = -m_p^2c^2 - 2m_pm_\pi c^2 - m_\pi^2c^2, \tag{2.19}$$

where the resulting particles only have rest energies $E = mc^2$ at threshold. The mass of any pion is denoted m_π.

Due to conservation and invariance of the squared four-momentum of the system, Eq. (2.15) equals Eq. (2.19), and with $p_p = \frac{E_p v_p}{c^2}$ turns into

$$-m_p^2c^2 - 2\frac{E_pE_\gamma}{c^2} - 2\frac{E_\gamma}{c}p_p = -m_p^2c^2 - 2m_pm_\pi c^2 - m_\pi^2c^2. \tag{2.20}$$

By cleaning and rearranging the equation above, the expression is reduced to

$$2E_pE_\gamma(1 + \frac{v_p}{c}) = (2m_pm_\pi + m_\pi^2)c^4, \tag{2.21}$$

which describes the kinematic threshold for a photomeson process where a neutral pion is produced. Assuming ultra-relativistic protons, $\beta_p = \frac{v_p}{c} \approx 1$, and again using $E_p = m_pc^2$ in the proton's rest frame, Eq. (2.21) reveals that for the photomeson process to occur, a photon in this frame needs energy greater than $m_\pi c^2(1 + \frac{m_\pi}{2m_p}) \approx 145$ MeV.

2.3.2 Bethe-Heitler pair production

Pair production, however, generally refers to the interaction between a photon and a nucleus, creating a new subatomic particle and its antiparticle. One particular type of pair production is the Bethe-Heitler process, which is the production of an electron and its antiparticle counterpart – a positron – described by $p + \gamma \rightarrow p + e^- + e^+$. For the rest of the thesis, pair production refers to this proton-photon interaction, unless otherwise specified.

Similar to the photomeson production, pair production is also defined by a threshold. At this threshold, the incoming photon, in the rest frame of a proton, needs energy greater than the rest energy of the electron and positron combined (the notation e^+ and e^- is excluded when referring to its masses). Summing up each rest energy and using conservation laws, consequently, a photon with energy $E_\gamma = 2m_ec^2 \approx 1$ Mev in the rest frame of the incoming proton is just enough to create such a pair. It is clear that 1 MeV for pair production is much lower than the 145 MeV for photomeson production. Other conservation laws which determine whether a particle process is valid, are, to mention a few, angular momentum, charge, baryon number and lepton number. These are all conserved when considering electron-positron or pion production from a proton-photon interaction.

Electrons as a direct result of pair production, and secondary outcome of photomeson production, can "cool" down through what is called synchrotron radiation (see

e.g. [1]). Hence, a key element in understanding both of these particle productions is the mechanisms of such radiation, since it inversely gives information about the initial proton. For that reason, the next subsection have a look at synchrotron radiation in general, and later examine how it is implemented in `agnpy`.

2.3.3 Synchrotron emission

Synchrotron emission occurs when the path of a relativistic charged particle is bent by a magnetic field (see e.g. [4]). This phenomenon is closely related to bremsstrahlung, but must not directly be confused with each other. Bremsstrahlung describes the resulting electromagnetic radiation emitted by any decelerated charged particle in the presence of another charge or Coulomb field – hence, the name braking radiation. Synchrotron radiation, however, specifically refers to a deceleration perpendicular to the initial charged particles velocity in the presence of a magnetic field (see e.g. [5]). To clarify; synchrotron radiation is a special case of bremsstrahlung, and due to the condition of a magnetic field, often called magnetobremsstrahlung. In the case of a non-relativistic charged particle, this type of breaking radiation is called cyclotron radiation. Since many celestial sources, such as the jets of an AGN, have both magnetic fields and relativistic charged particles, synchrotron radiation becomes a relevant emission when looking at relativistic particle processes. We will in the next section examine the source code `agnpy` and how it has implemented this synchrotron radiation.

2.4 Agnpy

The publicly available open-source code `agnpy` houses many functions to calculate SEDs of different particle processes, and some accompanying tutorials on how to utilize them. Some of these processes and types of radiation are

- Synchrotron radiation
- Inverse Compton process
- Absorption by photon-photon pair production

Among these, there are also functionalities to initialize non-thermal particle energy distributions and emission regions (more specifically a blob). Nevertheless, the code has only a few dependencies: `numpy`, `scipy` and `matplotlib` are familiar to many, and `sherpa` and `gammapy` are used in the special case of using `agnpy` for fitting. Finally, and especially important for this thesis and astrophysical modeling in general, is the package `astropy` [6, 7, 8]. This package covers much of the computational needs in astrophysics, but the most relevant feature for this project is the power to assign units to quantities, and to retrieve a multitude of physical constants. With this package, we have the capability to multiply, divide, and do all sorts of questionable mathematics to numbers and arrays and still wind up with the correct units. Highly practical!

As promised, we will now investigate how synchrotron emission is implemented in `agnpy`. The particular equation for synchrotron flux distribution stems from

Eq. (21) in Finke, Dermer and Böttcher (2008) [9], which again originates from Crusius and Schlickeiser (1986) [10]. A more detailed derivation, however, can be found in both Dermer and Menon (2009) [4] and Ghisellini (2013) [5]. Here, the derivations from the latter authors are recited to eventually end up at the equation implemented in `agnpy`.

The pitch-angle-averaged synchrotron spectral power of a single electron is given as Eq. (7.47) in [4], and reads

$$\langle P^{syn}(\nu) \rangle = \frac{\sqrt{3} e^3 B}{m_e c^2} \int_1^\infty d\gamma N_e(\gamma) R(x). \tag{2.22}$$

Here, B is the strength of the magnetic field in question, N_e is the electron distribution, and $R(x)$ is a function derived in [10]. An approximate version of $R(x)$ is written as

$$R(x) = \begin{cases} 1.80842 x^3, & x << 1 \\ \frac{\pi}{2} exp(-x)(1 - \frac{99}{162x}), & x >> 1. \end{cases} \tag{2.23}$$

The variable x is a function of several other values, namely $x = \frac{4\pi m_e c \nu}{3eB\gamma^2}$. By dimensional analysis, Eq. (2.22) has units of erg s^{-1} Hz^{-1}. The synchrotron flux $\nu F_\nu^{syn} = f_\epsilon^{syn}$ from a relativistic blob is then written as

$$f_\epsilon^{syn} = \frac{\delta_D^4 \nu' \left\langle P'^{syn}(\nu') \right\rangle}{4\pi d_L^2}, \tag{2.24}$$

where the primed quantities refer to the comoving frame. The factor of 4π in the denominator is a result of integrating over an isotropically radiating blob (see e.g. [5]). δ_D, however, is the Doppler factor, and appears in Eq. (2.24) to the fourth power as a consequence of beaming. Finally, d_L is the luminosity distance.

An equation for the synchrotron flux can also be found in Finke, Dermer and Böttcher (2008) [9], but written as

$$f_\epsilon^{syn} = \frac{\delta_D^4}{4\pi d_L^2} \sqrt{3} e^3 B \frac{\epsilon'}{h} \int_1^\infty d\gamma' N_e'(\gamma') R(x), \tag{2.25}$$

with $x = \frac{4\pi m_e^2 c^2 \epsilon'}{3eBh\gamma'^2}$. The difference between the two lies in the shift from ν' to ϵ'. The emitted photon's dimensionless energy in an observer's frame is denoted ϵ, and is therefore connected to the frequency by $\epsilon = \frac{h\nu}{m_e c^2}$. In the comoving frame, however, the same relation is given simply as $\epsilon' = \frac{h\nu'}{m_e c^2}$. This relationship shows the connection between the synchrotron flux described by Eqs. (2.24) and (2.25) found in [4] and [9], respectively.

However, one process which is not included in `agnpy` is the aforementioned pair production. Therefore, the end goal of this thesis is to suggest a functionality to calculate the SED of such a process – much like how the synchrotron emission is implemented, but now based on the formalism presented in [1]. The next chapter will deal with this theoretical framework, and later present a translation into numerical language.

CHAPTER

THREE

METHODS

3.1 Photo-pair production implementation

The energy distribution of pairs produced in the Bethe-Heitler process reads

$$\frac{dN}{dE_e} = \frac{1}{2\gamma_p^3} \int_{\frac{(\gamma_p + E_e)^2}{4\gamma_p^2 E_e}}^{\infty} d\epsilon \frac{f_{\rm ph}(\epsilon)}{\epsilon^2} \int_{\frac{(\gamma_p + E_e)^2}{2\gamma_p E_e}}^{2\gamma_p \epsilon} d\omega\omega \int_{\frac{\gamma_p^2 + E_e^2}{2\gamma_p E_e}}^{\omega - 1} \frac{dE_-}{p_-} W\left(\omega, E_-, \xi\right). \tag{3.1}$$

In the laboratory frame, ϵ denotes the photon energy, and E_e is the electron energy, in units of $m_e c^2$. The familiar proton Lorentz factor is, as usual, denoted γ_p. The target photon distribution is written as $f_{ph}(\epsilon)$.

In the proton's rest frame, however, the photon and electron energy are denoted ω and E_-, respectively, both in units of $m_e c^2$. The electron momentum is also marked with the same subscript, here p_-. $W(\omega, E_-, \xi)$ can be found in Blumenthal (1970) [11] and describes the differential cross section in the same frame. With all of this, Eq. (3.1) describes the energy distribution of electrons in the laboratory frame, and does not distinguish between electrons and positrons. But before a numerical adaptation is suggested, the integration limits deserve a thorough treatment.

3.1.1 Integration limits

Let us first take a look at the last integral which concerns the electron in the rest frame of the proton. To begin with, the electron four-momentum in the laboratory frame is denoted $\vec{P_e} = (\frac{E_e}{c}, p_x, p_y, 0)$. Thus, by connecting the laboratory frame with the proton's rest frame with an inverse Lorentz boost, as described by Eq. (2.8), we are left with

$$\frac{E_e}{c} = \gamma_p (\frac{E_-}{c} + \beta_p \frac{p_-}{c} \cos(\theta_-)). \tag{3.2}$$

From this, we assume ultra relativistic protons, $\beta_p \approx 1$, and that the two constituents are in line, $\theta_- = 0$. These assumptions give

$$E_e = \gamma_p (E_- + p_-). \tag{3.3}$$

From here, the energy-momentum relation described by Eq. (2.9) (which in units of $m = c = 1$ can be written as $E_-^2 = p_-^2 + 1$), is used to rewrite Eq. (3.3) into

$$E_e = \gamma_p(E_- + \sqrt{E_-^2 - 1}). \tag{3.4}$$

Rearranging this equation and squaring both sides yields

$$(E_e - \gamma_p E_-)^2 = \gamma_p^2(E_-^2 - 1). \tag{3.5}$$

By expanding the left square and rearranging some more, the electron energy in the proton's rest frame can be expressed as

$$E_- = \frac{E_e^2 + \gamma_p^2}{2\gamma_p E_e}, \tag{3.6}$$

which is the lower limit in the last integral. To get an upper energy limit, however, we need to regard the situation where all of the energy went into the electron, instead of being distributed among both an electron and a positron. In that case, the positron carries no additional kinetic energy. Nevertheless, it always possesses rest energy $E = m_e c^2$, which sets an upper limit of how much energy is actually transferred to the electron in the proton's rest frame. Given units of $m_e c^2$, this energy transfer can be written as $\omega = E_- + 1$, where ω denotes the energy of a photon in the rest frame of a proton with the same units. This simple expression is rearranged to get an upper limit of the electron energy as such

$$E_- = \omega - 1. \tag{3.7}$$

Next, consider the limits of the second integral. Using what is just obtained for E_-, along with Eq. (3.7), ω can be written as

$$\omega = E_- + 1 = \frac{E_e^2 + \gamma_p^2}{2\gamma_p E_e} + \frac{2\gamma_p E_e}{2\gamma_p E_e} = \frac{(E_e + \gamma_p)^2}{2\gamma_p E_e}, \tag{3.8}$$

which is the lower integration limit, again with units of $m_e c^2$. Eq. (61) from [1], $\omega = (u_p \cdot k) = \epsilon\gamma_p(1 - \cos(\theta))$, is used to obtain the upper integration limit. The largest value of ω is when $\theta = \pi$. Therefore,

$$\omega = \epsilon\gamma_p(1 - (-1)) = 2\epsilon\gamma_p. \tag{3.9}$$

Finally, the lower limit of the first integral is found when inserting Eq. (3.8) into Eq. (3.9), yielding

$$\epsilon = \frac{\omega}{2\gamma_p} = \frac{(E_e + \gamma_p)^2}{4\gamma_p^2 E_e}. \tag{3.10}$$

As there exists no upper restriction for the allowed photon energy, the upper theoretical integration limit is set to infinity.

3.1.2 Cross section

The differential cross section $W(\omega, E_-, \xi)$, where $\xi = \cos(\theta_-) = \frac{\gamma_p E_- - E_e}{\gamma_p p_-}$, is written as $\frac{d^2\sigma}{dE_- d\cos(\theta_-)}$. Following the notation in in Blumenthal (1970) [11], the entire expression can be written as

$$\begin{aligned}\frac{d^2\sigma}{dE_- d\cos(\theta_-)} = \left(\frac{\alpha Z^2 {r_0}^2 p_- p_+}{2k^3}\right)\Bigg[-4\sin^2(\theta_-)\frac{2{E_-}^2+1}{{p_-}^2{\Delta_-}^4} \\ + \frac{5{E_-}^2 - 2E_+E_- + 3}{{p_-}^2{\Delta_-}^2} + \frac{{p_-}^2 - k^2}{T^2{\Delta_-}^2} + \frac{2E_+}{{p_-}^2\Delta_-} + \frac{Y}{p_- p_+}\Bigg(2E_-\sin^2(\theta_-)\frac{3k + {p_-}^2 E_+}{{\Delta_-}^4} \\ + \frac{2{E_-}^2\left({E_-}^2 + {E_+}^2\right) - 7{E_-}^2 - 3E_+E_- - {E_+}^2 + 1}{{\Delta_-}^2} + \frac{k\left({E_-}^2 - E_-E_+ - 1\right)}{\Delta_-}\Bigg) \\ - \frac{{\delta_+}^T}{p_+ T}\left(\frac{2}{{\Delta_-}^2} - \frac{3k}{\Delta_-} - \frac{k\left({p_-}^2 - k^2\right)}{T^2\Delta_-}\right) - \frac{2y_+}{\Delta_-}\Bigg],\end{aligned} \tag{3.11}$$

where

$$\begin{aligned}&T = |\mathbf{k} - \mathbf{p}_-|\,, \quad Y = \left(2/{p_-}^2\right)\ln\left[\left(E_+E_- + p_+p_- + 1\right)/k\right], \\ &y_+ = p_+^{-1}\ln\left[\left(E_+ + p_+\right)/\left(E_+ - p_+\right)\right], \quad {\delta_+}^T = \ln\left[\left(T + p_+\right)/\left(T - p_+\right)\right].\end{aligned} \tag{3.12}$$

k is here used to describe the photon energy in the rest frame of the proton – or ω as previously used. E_- is the familiar electron energy in the same frame, and E_+ is the energy of the positron, both in units of $m_e c^2$. The momentum of the electron and positron is written as p_- and p_+, respectively. In the prefactor, α is the fine structure constant, Z is the nuclear charge, and r_0 is the classical electron radius. A rewritten version of the differential cross section above can also be found in [12].

3.1.3 Planckian distribution

In astrophysical environments, an important case of target photons is that of a Planckian distribution. This distribution, which in [1] is denoted Eq. (65), can be written as

$$f_{\rm ph}(\epsilon) = \frac{1}{\pi^2}\frac{\epsilon^2}{e^{\epsilon/k_B T} - 1}, \tag{3.13}$$

which in turn can be written as a differential as such

$$\frac{f_{ph}(\epsilon)}{\epsilon^2} = \frac{1}{\pi^2}\frac{e^{-\epsilon/k_B T}}{1 - e^{-\epsilon/k_B T}} = \frac{k_B T}{\pi^2}\frac{-e^{-\epsilon/k_B T}}{1 - e^{-\epsilon/k_B T}}\left(-\frac{1}{k_B T}\right) \tag{3.14}$$

$$\frac{f_{ph}(\epsilon)}{\epsilon^2} = \frac{k_B T}{\pi^2}\frac{d}{d\epsilon}\ln\left(1 - e^{-\epsilon/k_B T}\right) \tag{3.15}$$

$$d\epsilon\frac{f_{\rm ph}(\epsilon)}{\epsilon^2} = \frac{k_B T}{\pi^2} d\ln\left(1 - e^{-\epsilon/k_B T}\right). \tag{3.16}$$

Performing the integration over $d\epsilon$ by parts (see e.g. [1]), along with a transformation of the integral, here denoted Eq. (3.1), leads to

$$\frac{dN}{dE_e} = -\frac{k_B T}{2\pi^2 \gamma_p^3} \int_{\frac{(\gamma_p + E_e)^2}{2\gamma_p E_e}}^{\infty} d\omega \omega \ln\left(1 - e^{-\omega/(2\gamma_p k_B T)}\right) \int_{\frac{\gamma_p^2 + E_e^2}{2\gamma_p E_e}}^{\omega - 1} \frac{dE_-}{p_-} W\left(\omega, E_-, \xi\right). \tag{3.17}$$

The expression above describes the energy distribution of pairs from the Bethe-Heitler process, given a Planckian distribution of target photons. We will in the next section provide a method to calculate this expression numerically.

3.2 Numerical implementation

To begin with, an approach of `numpy` arrays is used to verify Eq. (3.16) numerically. We start by initializing an array for the photon energy, ϵ, which contains a set of energies ranging from $2.585 \cdot 10^{-5}$ to $2.585 \cdot 10^{-3}$ in units of electron volts (eV). Next, we define a function `Planckian_dist` which uses such an `epsilon_array` as a parameter and returns a Planckian distribution as in Eq. (3.13). The temperature T is here set to the cosmic microwave background (CMB) temperature of 2.7 K. The Boltzmann constant k_B, however, is obtained from the `astropy` package. An important note is that the product $k_B T$ will automatically be initialized in joules (J), where we want it in eV. Again, the genius `astropy` has the capacity to convert between the desired units. Finally, the two components left are the differentials $d\epsilon$ and $d\ln(1 - e^{\epsilon/k_B T})$. Numerically, these differentials are approximated with the literal spacing between each element in `epsilon_array` and the new logarithmic part, respectively. Consequently, these are both arrays themselves, but one element shorter. We naively add the last element again to acquire the correct lengths. Finally, we can compare the two sides of Eq. (3.16) to examine the integrity of this numerical implementation, shown in Fig. (3.2.1).

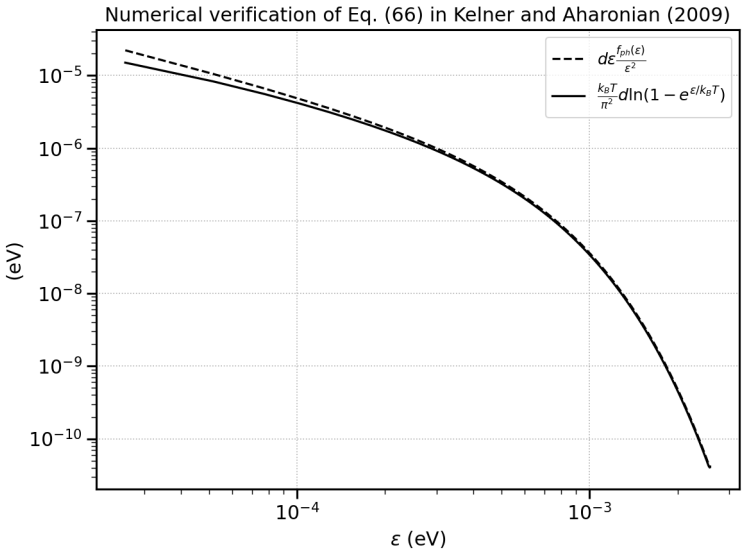

Figure 3.2.1: Plot comparing the the two sides of Eq. (3.16) as a function of ϵ. Here, $T = T_{CMB} = 2.7$ K (log-log plot).

The two sides coincide to a great degree, especially for higher photon energies.

Hence, one can not only justify the use of a Planckian distribution to modify Eq. (3.1) into Eq. (3.17), but we also know that the numerical approach of using arrays and approximating differentials is reasonable.

However, before embarking on the implementation of Eq. (3.17), we need to address one major concern: numerical integration. The integrands can be created with `numpy` arrays as done above, but the integration itself remains. Once again, `agnpy`, and more specifically `scipy` [13], comes to the rescue. This package contains functionalities for numerical integration, and we will make use of the trapezoidal method to be specific. With these tools, we are now ready to numerically calculate the energy distribution described by Eq. (3.17).

First and foremost, we fix the proton energy $E_p = 6.4 \cdot 10^{19}$ eV, and the Lorentz factor as $\gamma_p = E_p/m_p c^2$. The CMB temperature is set to 2.7 K. Then, we initialize an array of electron energies in the laboratory frame. We call this array `E_e_array` and fill it up with energies ranging from 10^{12} to 10^{23} eV. Every value in this array is divided by $m_e c^2$ to obtain the electron energy in dimensionless units. Now, we choose the first element in this array and use it in the lower integration limit of the first (and later second) integral. Since the total integral yields a single value for a single E_e, there is a one-to-one correspondence of the two components.

Then, for this E_e, we create an array of ω ranging from the dimensionless lower to upper integration limit. We call this array `omega_array`, and make sure it contains the same amount of points as the initial `E_e_array`. Bear in mind that the upper theoretical integration limit extends towards infinity. Numerically, this issue is resolved by simply fixing it as a huge number – say 10^7. How this impacts the overall integral will be addressed later on. Now, for one value of E_e in `E_e_array`, we have obtained a uniquely defined `omega_array` ranging from the lower to upper integration limit. Before going any further, we look at these integration limits as a function of the E_e in `E_e_array`. This relationship is illustrated in Fig. (3.2.2).

This plot shows the integration limits as functions of the electron energy. In other words, this is the first and last element in each `omega_array` created for a particular E_e in `E_e_array`. As expected, the upper limits are always constant, since it is defined as 10^7, regardless of the current E_e. Having said that, the minimum values show a more interesting shape. We clearly see a curved arc with a minimum around $E_e = 10^{17}$ eV. Remember, although E_e is plotted in eV, ω is still defined in units of $m_e c^2$, thus, dimensionless. In addition to this, bear in mind that the arrays themselves are not numerically shorter as the end points change. They still contain the same amount of points.

Next, we create a function which uses a unique `omega_array` and returns the logarithmic part of the first integrand. As `omega_array` is defined in units of $m_e c^2$, we need to make sure that the denominator of the exponent in $\ln\left(1 - e^{-\omega/(2\gamma_p k_B T)}\right)$ is defined in the same way. The resulting array is called `ln_array`.

To summarize, we have now created a unique `omega_array` and a corresponding `ln_array` for a single E_e. Next, for the first ω in this `omega_array`, we create an

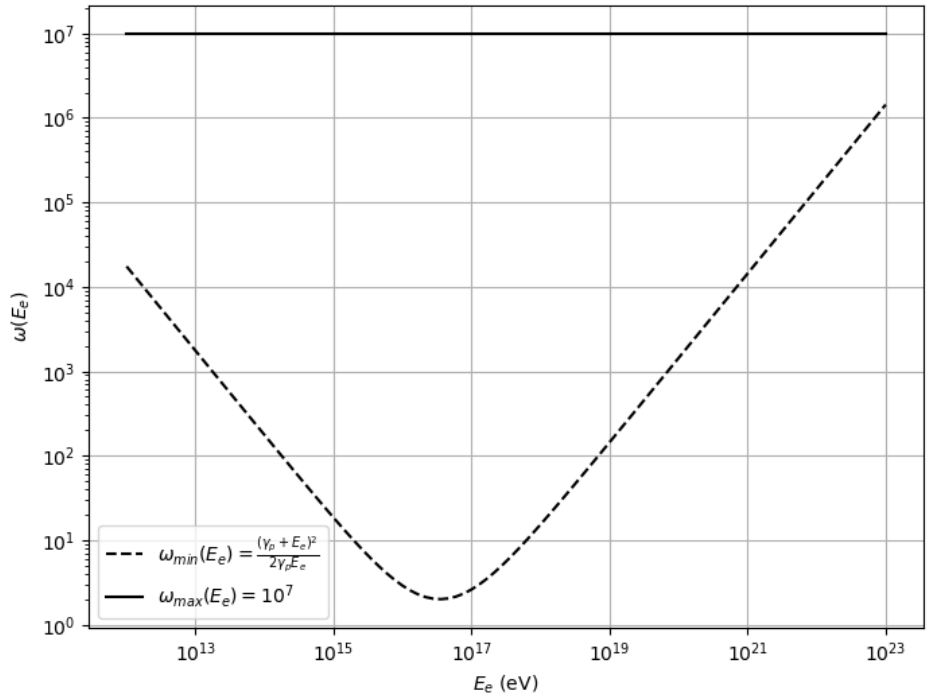

Figure 3.2.2: Plot showing the lower and upper integration limit of the first integral in Eq. (3.17) as a function of E_e. Here, $E_p = 6.4 \cdot 10^{19}$ eV. The results are in dimensionless units (log-log plot).

array of electron energies in the proton's rest frame, E_-, according to the integration limits of the *second* integral. This latter array is called `E_m_array`. Thus, if `E_e_array` contains N elements, we obtain N `omega_arrays` and N^2 `E_m_arrays`.

Then, from this `E_m_array`, we create the corresponding array of electron momentum in the proton's rest frame by $\beta_- E_-$ and name it `p_m_array` (formula obtained from the energy-momentum relation in units of mc^2 and natural units, $m = c = 1$). Finally, the differential cross section is created using `p_m_array`, `E_m_array`, the corresponding ω, and the initial E_e which started it all.

We have now reached the end of the total expression, and we can start backtracking the process. The second integral is calculated with `scipy` which yields a single value. Afterwards, we choose the next value of omega in the first `omega_array`, and calculate the integral again. Another value is returned. Eventually, we reach the end of our `omega_array`. What is left is an additional array containing the results of the second integral for each ω in `omega_array`. The integrand of the *first* integral is then `omega_array` $\times$ `ln_array` $\times$ `second_integral_array`. The three arrays are multiplied together to form the total first integrand. Once again, `scipy` is used to calculate this first integral. As expected, this yields a single value, since we started with a single E_e. After all that, we choose the next E_e in `E_e_array`, and the whole process is executed again.

CHAPTER

FOUR

RESULTS

The algorithm explained above is saved as a function itself. The only parameters are the initial proton Lorentz factor, γ_p, and an arbitrary array of electron energies, E_e. The function is now used for three different proton energies, $E_p = 6.4{\cdot}10^{19}, 10^{20}$ and $3 \cdot 10^{20}$ eV, and an electron energy array ranging from 10^{12} to 10^{23} eV. A plot of the resulting SED of electrons from the pair production process is shown in Fig. (4.0.1).

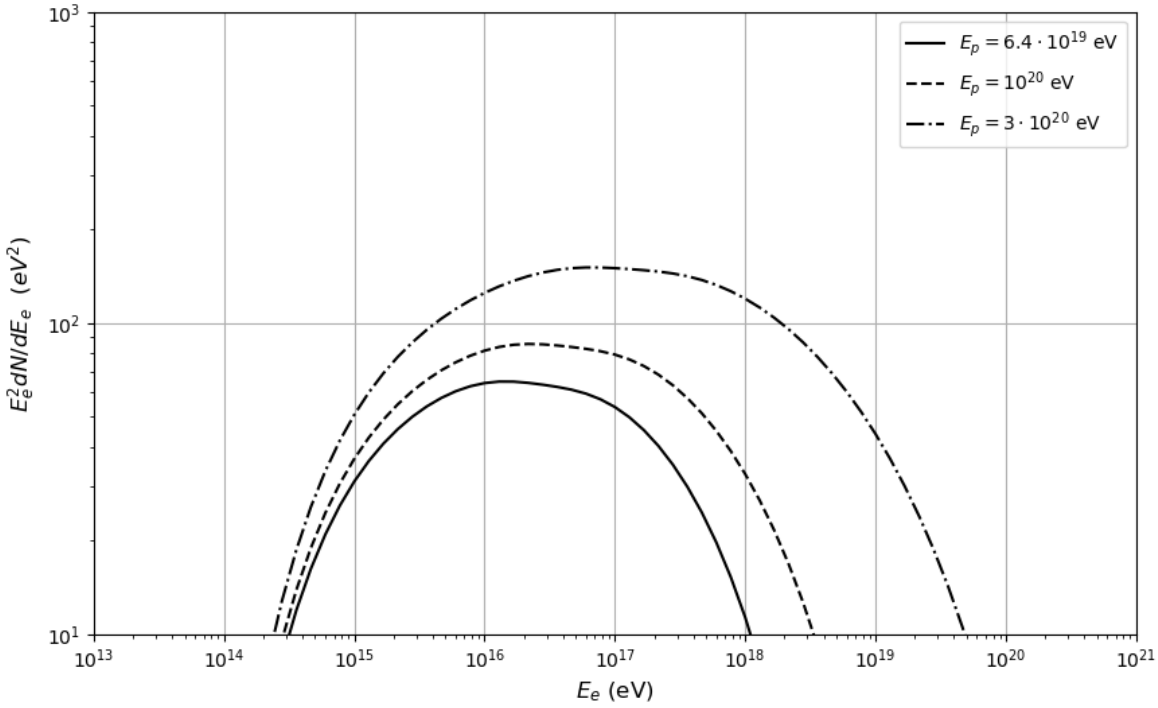

Figure 4.0.1: Plot showing the SED produced by the code as a function of E_e for three different proton energies, multiplied by a scaling factor of $1.4 \cdot 10^{11}$ (see text for further explanation). Every parameter is defined in units of $m_e c^2$, thus, dN/dE_e is also in dimensionless units (log-log plot).

The whole SED is divided by r_0^2 to cancel out the area dependence from the differential cross section. Every parameter of the integral (including $k_B T$ in the

prefactor) is defined in units of $m_e c^2$. The resulting dN/dE_e is therefore in dimensionless units, even though E_e is plotted in eV. In addition to this, the axis limits are defined in accordance with Fig. (10) in [1] for comparable results. Regardless, the resulting SED deviates from the one in [1] by a factor of approximately $1.4 \cdot 10^{11}$, probably originating from a missing constant or unit conversion. Nevertheless, given the time frame of this thesis, we leave the investigation of the missing ingredient as future work. The scaling factor is included in the figure above to display similar scales, and we can see that our code produces SEDs of electrons that closely resembles the ones in [1].

For each value of E_p, one distinct bell shaped curve of $E_e^2 dN/dE_e$ is produced. Each curve starts off around $E_e \approx 10^{14}$ eV, and for increasing proton energies, grows larger with a maximum around $E_e \approx 10^{-3} E_p$, still keeping the well defined bell shape. Even though the starting point of each curve is about the same, the SED is seen to span a larger range of electron energies for increasing proton energies. Having said that, the maxima of each curve do not deviate from one another by even one order of magnitude.

From [1], it is expected that the similar SED of electrons from the photomeson process, however, will appear several orders of magnitude higher for increasing proton energies. In turn, the combination of the SED from both processes will make it hard to recognize the signature of pair production, since it is overshadowed by the taller photomeson curve. In addition to this, we expect the latter curve to appear with a more slanted bell shape, spanning a smaller range of electron energies, and to the right of the pair production curve.

For demonstrative purposes, plots showing the shape of the cross section for four different values of E_e, and four different ω in the corresponding `omega_arrays`, are shown in Fig. (4.0.2). We have used the fact that $\omega = E_- + E_+$ to account for the energy of the positron in the same frame.

From these four plots, we can clearly see how much the differential cross section changes for incremental modifications of its parameters. Remember that one unique `omega_array` is created for a single E_e, and we have chosen the same four indices of the `omega_arrays` to obtain four corresponding values of ω. One of these, again, increases the upper limit in `E_m_array`, which is the reason why the differential cross section is seen to span a larger range of E_- as ω grows. The upper left plot displays the low-energy cutoff, whereas the high-energy equivalent is more clearly defined for a larger value of E_e. We can also see that the magnitude steadily decreases for increasing E_e. We use this differential cross section from [11] instead of the approximation in [4] (derived from [14]), since the latter is already integrated over the electron energy. Consequently, the information of E_- is lost, and the expression is therefore not suitable for our purposes. Implementing the differential cross section from [11] is tedious, but rewarding nonetheless.

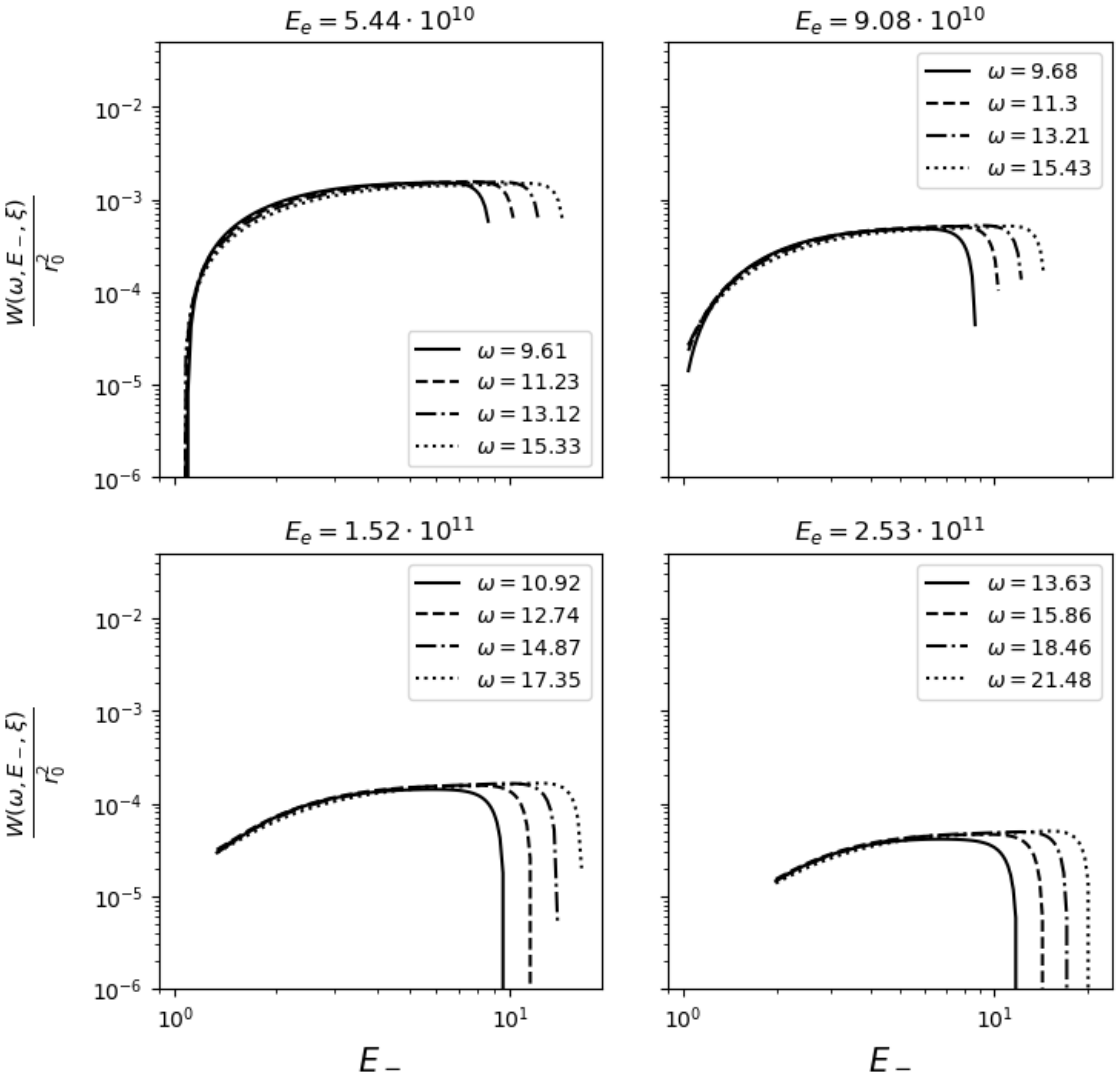

Figure 4.0.2: Plots showing the differential cross section as a function of E_- for four different E_e and four different ω in the corresponding `omega_arrays`. Here, $E_p = 6.4 \cdot 10^{19}$ eV for every plot. Every parameter is defined in units of $m_e c^2$ (log-log plot).

CHAPTER

FIVE

DISCUSSION

5.1 Efficiency of the code

First and foremost, it is worth mentioning that the algorithm is fairly slow. Recall that 100 values of E_e leads to 100 iterations in the first for-loop. Each of these requires another 100 iterations for the second integral, assuming that every array is created with the same amount of points. Initially, 100×100 iterations is not necessarily a challenge for a small-scale `python` program, but one can imagine why the code struggles by taking a glance at the comprehensive cross section. In spite of that, there are measures to accelerate the algorithm by a considerable amount. One such measure is the JIT compiler `numba` [15]. However, using `numba` requires us to discard the use of some imported functionalities. For instance, `scipy` would have to be abandoned in favor of the built-in integration method in `numpy`, which, in turn, limits the scale of the code since the only integration method in `numpy` is the trapezoidal method. In addition to this, one would have to be careful when using the handy functionalities of `agnpy`, as the JIT compiler does not generally prefer important packages. Nevertheless, a fast code would undoubtedly be a huge advantage.

5.2 Upper integration limit

We return now to the upper limit of the first integral, which in our code is set to a fixed value of 10^7. Initially, one would assume that an even higher number would give more accurate results, and, in turn, affect the efficiency of the code. This hypothesis to put to the test by running the code for four increasing upper integration limits, presented in Fig. (5.2.1).

It is very clear from the plot that an increase in the upper integration limit gives insignificant – if any – change in the overall SED. This, in turn, indicates that there is a ceiling where no more accuracy is gained by going beyond. For this ceiling-value, the integral stabilizes – converges – and there is no reason to over-exaggerate the upper integration limit when a smaller number will suffice. To find this ceiling, however, is left as future work.

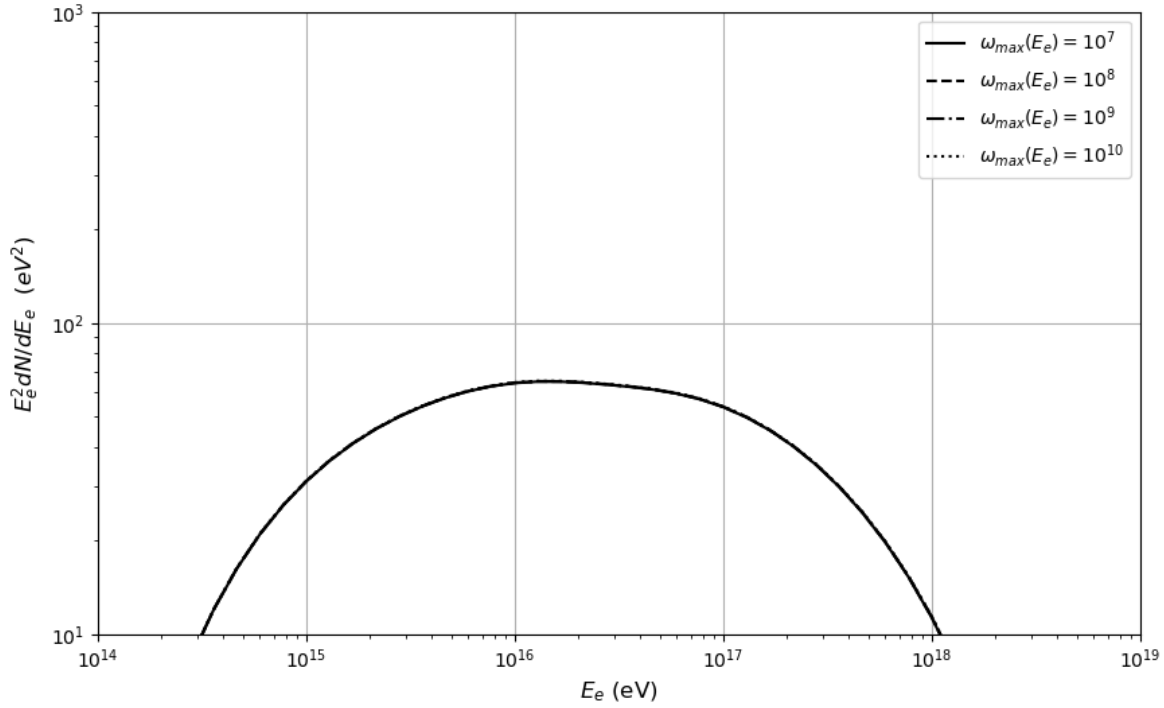

Figure 5.2.1: Plot showing the SED produced by our code as a function of E_e for four increasing values of ω_{max} (upper integration limit), multiplied by a scaling factor of $1.4 \cdot 10^{11}$. Here, $E_p = 6.4 \cdot 10^{19}$ eV (log-log plot).

Some issues may arise, though, when the upper limit draws near the highest value of the *lower* integration limit – or even further below. From Fig. (3.2.2), this would mean that the straight line moves down – even crosses – the line that represents the lower limit. As a consequence, for some `omega_arrays`, the first element is actually larger than the last element, hence, a decreasing nature of the array. This is unproblematic at first, but may give rise to negative values when integrating. Negative values would, of course, not appear in a log-log plot. Again looking at Fig. (3.2.2), as the straight line moves further and further down, more and more `omega_arrays` will have this decreasing nature, starting off with just the last ones, then accompanied by the first ones, and finally, everyone. As a consequence, if we decrease the upper integration limit substantially, we could expect the SED to become "slimmer" (since negative values are not shown), starting from the right, then from both sides, and finally, disappear. So, to avoid this issue all together, we could make sure that the upper integration limit is dependent on the lower one – say e.g. $\omega_{max} = 10^5 \omega_{min}$.

5.3 Future work

Since the pair production process is not directly involved in the creation of neutrinos, it is often an overlooked process in high-energy astrophysics. It also serves as a subdominant proton cooling process, especially for protons with energies that exceed the threshold for the photomeson processes (see e.g. [16]). As previously shown, this threshold lies at energies close to 145 MeV. Still regarding the under-

appreciated nature of pair production, Petropoulou and Mastichiadis (2014) [16] entertain the idea that pair production may, in fact, leave a noticeable mark on the SED of blazars.

The theoretical model used in [16] is that of a spherically radiating blazar initialized as a blob (remember that creating radiating blobs is one of the many functionalities in `agnpy`). Also assumed are power-law distributions for interacting electrons and protons. From their model parameters and assumptions, it is theorized that the signature of pair production will appear as a third "hump" in a SED plot of a blazar; between the left and right arc showing synchrotron emission from primary electrons and photomeson process, respectively. The theoretical framework is then developed into a numerical code, which is used to computationally verify this speculation. Consequently, the results derived in [16] show that pair production "fills the gap between the low- and high energy components of the SED" (Petroploulou and Mastichiadis, 2014, p. 46), which, to some degree, justifies this otherwise glossed-over particle process.

Fig. (5.3.1) is an adaptation of Fig. (4) in [16], and shows the electron synchrotron radiation from the Bethe-Heitler pair production process (pe) between SEDs from other processes: synchrotron self-Compton scattering (SSC), photopion production (pπ) and proton synchrotron radiation.

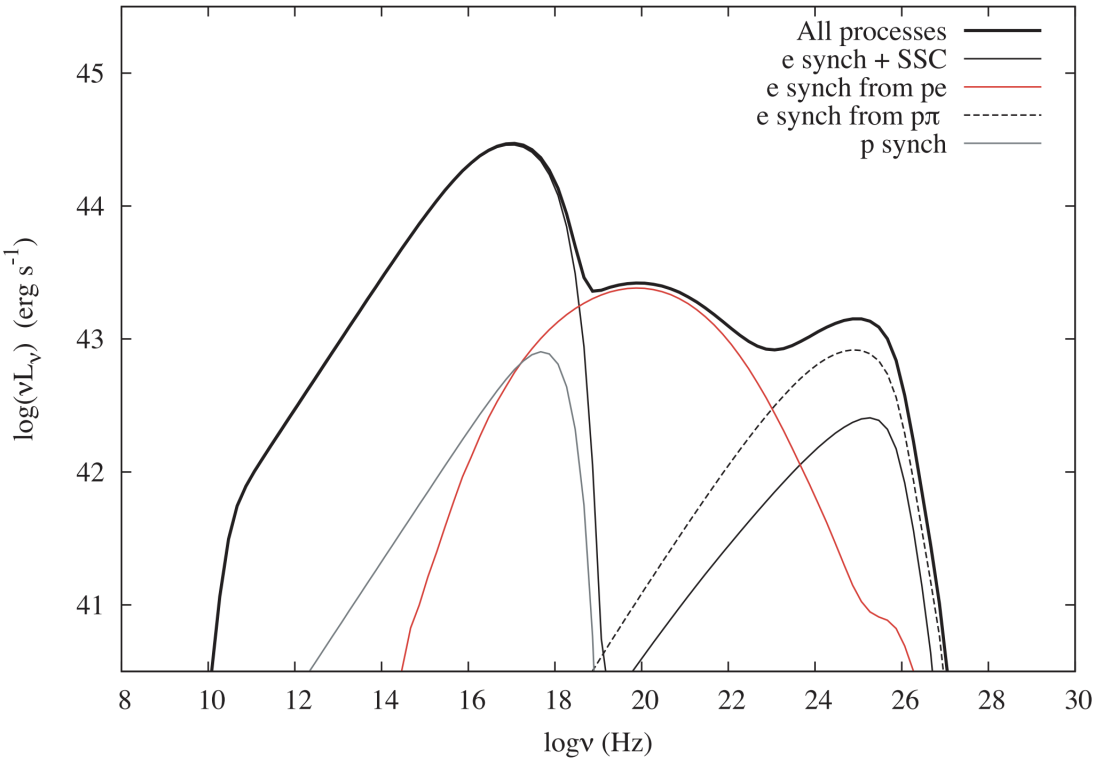

Figure 5.3.1: SEDs for various particle processes. Reprinted from "Bethe–Heitler emission in BL Lacs: filling the gap between X-rays and γ-rays," by M. Petropoulou and A. Mastichiadis, 2014, *Monthly Notices of the Royal Astronomical Society* 447.1, p. 43. Copyright 2014, Oxford University Press.

However, an important point to stress, is the fact that the signature of pair produc-

tion has yet to make its appearance in observed data. Nevertheless, the possibility to access the high-energy regime through future breakthroughs in telescope and detector technology could yield the sought-after information necessary to confirm this "hump". Petropoulou and Mastichiadis (2014) [16] mention *NuSTAR* and *INTERGAL* as two current missions which explore the regime of hard X-rays and soft γ-rays. Another reference is to the future satellite named *PANGU*, which will focus on γ-ray observations with high-sensitivity. Thus, in accordance with model projections and planned missions, the future for pair production, and high-energy astrophysics in general, is most definitely a bright one.

The future for this particular project, however, is perhaps more straightforward. The natural next step would be to suggest similar numerical implementations of other particle processes, which, in turn, can be used to further expand `agnpy`. A worthy candidate is of course the aforementioned photomeson process. Other processes of interest would be those directly involved in neutrino production. This Bachelor's thesis could perhaps develop into a Master's degree dealing with this very concept. The more the merrier!

Also interesting, regarding the specific results obtained here, would be to replace the initial proton with a whole distribution. A power-law distribution would be of particular relevance. By means of this, a multidimensional integration with γ_p as an array instead of a fixed value is required. Such a power-law distribution is taken into account in Figs. (18) and (19) in [1]. Obtaining similar results, but now utilizing `agnpy` and its packages, would be of great interest. Also, with the pair production process safely implemented in this source code, one could follow in the footsteps of [16] and give an additional numerical verification of the proposed "hump".

CHAPTER

SIX

CONCLUSIONS

This thesis aimed to calculate the SED of electrons from the Bethe-Heitler pair production process. Based on the formalism in [1] with a Planckian distribution of target photons, we present three such SEDs for three different proton energies. We see that our approach of `numpy` arrays, for-loops, and `agnpy` packages, gives distributions that are consistent with the broad bell shape distinctive to this particular process, and in excellent agreement with the results in [1]. Figuring out the scaling disagreement would of course make our functionality even more credible, but, given the time constraint of this thesis, left as future work.

In addition to this, we have derived the threshold energy for both the pair production and photomeson process, and examined how synchrotron emission is implemented in `angpy`. We have also analyzed the integration limits of the functional form of the SED from the pair production process, and discussed how to numerically consider an infinite upper integration limit. We present the differential cross section from [11] for different parameters, and while the implementation of this cross section is exhaustive, we find it absolutely necessary nonetheless, since it accounts for all of the variables of the integral. Even so, the many terms and iterations make the algorithm fairly slow, and we discuss potential measures that can increase the efficiency of the code.

Still, given the few parameters and dependencies, we believe the functionality presented here offers a satisfactory way of calculating the SED from the process in question. In addition to this, since our approach uses the same convention as `agnpy`, it could make a fine addition to the collection of similar functionalities in this source code, and would be a practical tool for others to use or even further build upon. However, our code is still a work in progress, and we would have to locate the missing scaling factor and make sure that the syntax is made as efficient as possible before submitting anything to this publicly available source code.

REFERENCES

[1] S. R. Kelner and F. A. Aharonian. "Energy spectra of gamma rays, electrons, and neutrinos produced at interactions of relativistic protons with low energy radiation". In: *Physical Review D* 78.3 (Aug. 2008). DOI: 10.1103/physrevd.78.034013. URL: https://doi.org/10.1103%5C%2Fphysrevd.78.034013.

[2] C. Nigro et al. "agnpy: An open-source python package modelling the radiative processes of jetted active galactic nuclei". In: 660, A18 (Apr. 2022), A18. DOI: 10.1051/0004-6361/202142000. arXiv: 2112.14573 [astro-ph.IM].

[3] *Introduction to active galaxies. An OpenLearn chunk used by permission of The Open Univerisy copyright © (2023).* URL: www.open.edu/openlearn..

[4] Charles D. Dermer and Govind Menon. *High energy radiation from black holes: Gamma rays, cosmic rays and neutrinos*. Princeton, USA: Princeton U. Pr., 2009.

[5] Gabriele Ghisellini. *Radiative Processes in High Energy Astrophysics*. Springer International Publishing, 2013. DOI: 10.1007/978-3-319-00612-3. URL: https://doi.org/10.1007%5C%2F978-3-319-00612-3.

[6] Astropy Collaboration et al. "Astropy: A community Python package for astronomy". In: 558, A33 (Oct. 2013), A33. DOI: 10.1051/0004-6361/201322068. arXiv: 1307.6212 [astro-ph.IM].

[7] Astropy Collaboration et al. "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". In: 156.3, 123 (Sept. 2018), p. 123. DOI: 10.3847/1538-3881/aabc4f. arXiv: 1801.02634 [astro-ph.IM].

[8] Astropy Collaboration et al. "The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package". In: *apj* 935.2, 167 (Aug. 2022), p. 167. DOI: 10.3847/1538-4357/ac7c74. arXiv: 2206.14220 [astro-ph.IM].

[9] Justin D. Finke, Charles D. Dermer, and Markus Böttcher. "Synchrotron Self-Compton Analysis of TeV X-Ray-Selected BL Lacertae Objects". In: *The Astrophysical Journal* 686.1 (Oct. 2008), p. 181. DOI: 10.1086/590900. URL: https://dx.doi.org/10.1086/590900.

[10] A. Crusius and R. Schlickeiser. "Synchrotron radiation in random magnetic fields". In: *AAP* 164.2 (Aug. 1986), pp. L16–L18.

[11] George R. Blumenthal. "Energy Loss of High-Energy Cosmic Rays in Pair-Producing Collisions with Ambient Photons". In: *Phys. Rev. D* 1 (6 Mar. 1970), pp. 1596–1602. DOI: 10.1103/PhysRevD.1.1596. URL: https://link.aps.org/doi/10.1103/PhysRevD.1.1596.

[12] Bruno Jiménez Fernández and Hendrik van Eerten. *Katu: a fast open-source full lepto-hadronic kinetic code suitable for Bayesian inference modelling of blazars*. 2021. arXiv: 2104.08207 [astro-ph.HE].

[13] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[14] Susan Stepney and P. W. Guilbert. "Numerical fits to important rates in high temperature astrophysical plasmas". In: *Monthly Notices of the Royal Astronomical Society* 204.4 (Oct. 1983), pp. 1269–1277. ISSN: 0035-8711. DOI: 10.1093/mnras/204.4.1269. eprint: https://academic.oup.com/mnras/article-pdf/204/4/1269/3155787/mnras204-1269.pdf. URL: https://doi.org/10.1093/mnras/204.4.1269.

[15] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. "Numba: A LLVM-Based Python JIT Compiler". In: LLVM '15. Austin, Texas: Association for Computing Machinery, 2015. ISBN: 9781450340052. DOI: 10.1145/2833157.2833162. URL: https://doi.org/10.1145/2833157.2833162.

[16] M. Petropoulou and A. Mastichiadis. "Bethe–Heitler emission in BL Lacs: filling the gap between X-rays and γ-rays". In: *Monthly Notices of the Royal Astronomical Society* 447.1 (Dec. 2014), pp. 36–48. ISSN: 0035-8711. DOI: 10.1093/mnras/stu2364. eprint: https://academic.oup.com/mnras/article-pdf/447/1/36/4895607/stu2364.pdf. URL: https://doi.org/10.1093/mnras/stu2364.

APPENDICES

A - GITHUB REPOSITORY

All codes used in this document are included in the Github repository linked below. Further explanations are given in the README-file.

Github repository link

- `https://github.com/andreasnossum/Bethe-Heitler-emission.git`