Hazma Documentation

Release 1.0

Logan A. Morrison et. al.

Dec 20, 2017

CONTENTS

1 Description **3**
- 1.1 Introduction . . . 3
- 1.2 Decay spectra . . . 3
- 1.3 Final State Radiation . . . 5

2 Modules **9**
- 2.1 Gamma Ray (`hazma.gamma_ray`) . . . 9
- 2.2 Muon (`hazma.muon`) . . . 10
- 2.3 Electron (`hazma.electron`) . . . 11
- 2.4 Charged Pion (`hazma.charged_pion`) . . . 12
- 2.5 Neutral Pion (`hazma.neutral_pion`) . . . 12
- 2.6 Charged Kaon (`hazma.charged_kaon`) . . . 13
- 2.7 Long Kaon (`hazma.long_kaon`) . . . 13
- 2.8 Short Kaon (`hazma.short_kaon`) . . . 13

3 Usage **15**

4 Indices and tables **17**

Python Module Index **19**

Index **21**

Contents:

CHAPTER

ONE

DESCRIPTION

1.1 Introduction

This package is used to compute the gamma ray spectra $\frac{dN}{dE}$ for light particles, such as, pions, kaon, electrons and muons, in an energy regime where the mass effects are important, i.e. is the MeV energy range. The code has been written in python/cython.

1.2 Decay spectra

In this section, we describe how the radiative decay spectra are computed for the muon, charged pion and neutral pion.

1.2.1 Muon

The dominant contribution to the radiative decay of the muon comes from $\mu^{\pm} \to e^{\pm}\nu\bar{\nu}\gamma$. The unpolarized differential branching fraction of this decay mode in the *muon rest frame* can be written as [1]

$$\frac{dB}{dy\, d\cos\theta_{\gamma}^{R}} = \frac{1}{y}\frac{\alpha}{72\pi}(1-y)\left[12\left(3-2y(1-y)^2\right)\log\left(\frac{1-y}{r}\right)+y(1-y)(46-55y)-102\right]$$

where $r = (m_e/m_{\mu})^2$, $0 \leq y = 2E_{\gamma}^{R\mu}/m_{\mu} \leq 1-r$, ($E_{\gamma}^{R\mu}$ is the energy of the photon in the muon rest frame) and θ_{γ}^{R} is the angle the photon makes with respect to some axis in the muon rest frame. In order to obtain the decay spectrum in the laboratory frame, we need to boost the above spectrum. In other words, we need to change variables from the gamma ray energy and angle in the muon rest frame to those in the lab frame. We then integrate over the angle to compute dN/dE_{γ}. The Jacobian for this change of variables is

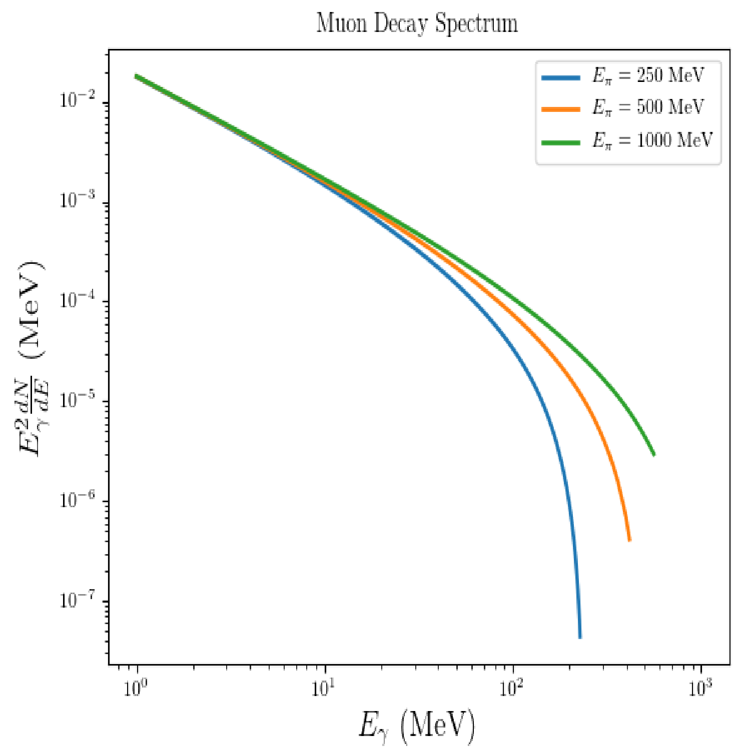
$$J = \frac{1}{2\gamma(1-\beta\cos\theta_{\gamma}^{L})}$$

where the boost parameters are

$$\gamma = E_{\mu}/m_{\mu}, \qquad \beta = \sqrt{1-\left(\frac{m_{\mu}}{E_{\mu}}\right)^2}$$

Integrating over angles yields the gamma ray spectrum in the lab frame:

$$\frac{dN}{dE_{\gamma}^{L}} = \int_{-1}^{1} d\cos\theta_{\gamma}^{L}\frac{1}{2\gamma(1-\beta\cos\theta_{\gamma}^{L})}\frac{dB}{dE_{\gamma}^{R\mu}}$$



1.2.2 Charged Pion

To compute the gamma ray spectrum from a charged pion, one considers to possible decay modes. These decay modes are $\pi^{\pm} \to \mu^{\pm}\nu_{\mu}\gamma$ and $\pi^{\pm} \to \mu^{\pm}\nu_{\mu} \to e^{\pm}\nu_{\mu}\nu_{\mu}\nu_{e}\gamma$. To compute the gamma ray spectrum from the first decay mode, one uses results from [2]. It turns out that the spectrum from this decay mode is roughly a factor of 100 times smaller than the spectrum from the second decay mode. We thus ignore the contributions from $\pi^{\pm} \to \mu^{\pm}\nu_{\mu}\gamma$.

To compute the γ-ray spectrum from $\pi^{\pm} \to \mu^{\pm}\nu_{\mu} \to e^{\pm}\nu_{\mu}\nu_{\mu}\nu_{e}\gamma$, we first take the muon decay spectra (see section on muon decay spectra) and boost the muon into the pion rest frame use the following:

$$\gamma_1 = E_{R\mu}/m_{\mu} \qquad \beta_1 = \sqrt{1 - \left(\frac{m_{\mu}}{E_{R\mu}}\right)^2} \qquad E_{R\mu} = \frac{m_{\pi}^2 - m_{\mu}^2}{m_{\pi}^2 + m_{\mu}^2}$$

where $E_{R\mu}$ is the energy of the muon in the pion rest frame. The photon spectrum in the charged pion rest frame, $dN/dE_{\gamma}^{R\pi}$, is obtain by integrating the differential branching ratio times a Jacobian factor $1/2\gamma_1(1 - \beta_1 \cos\theta_{\gamma}^{R\pi})$ over the angle the photon makes with the muon. Once this integration is completed, one then boosts into the laboratory frame of reference. The steps are nearly identical to boosting from the muon rest frame to the pion rest frame. The only thing that changes in the boost factor and the Jacobian. In going from the charged pion rest frame to the laboratory frame, the Jacobian and boost factor are

$$J = \frac{1}{2\gamma_2(1 - \beta_2 \cos\theta_{\gamma}^{L})} \qquad \gamma_2 = E_{\pi}/m_{\pi} \qquad \beta_2 = \sqrt{1 - \left(\frac{m_{\mu}}{E_{\pi}}\right)^2}$$

The gamma-ray spectrum in the laboratory frame will thus be

$$\frac{dN}{dE_{\gamma}^{L}} = \int_{-1}^{1} d\cos\theta_{\gamma}^{L} \frac{1}{2\gamma_2(1 - \beta_2\cos\theta_{\gamma}^{L})} \times \left(\int_{-1}^{1} d\cos\theta_{\gamma}^{R\pi} \frac{1}{2\gamma_1(1 - \beta_1\cos\theta_{\gamma}^{L})} \frac{dB}{dE_{\gamma}^{R\mu}}\right)$$

where

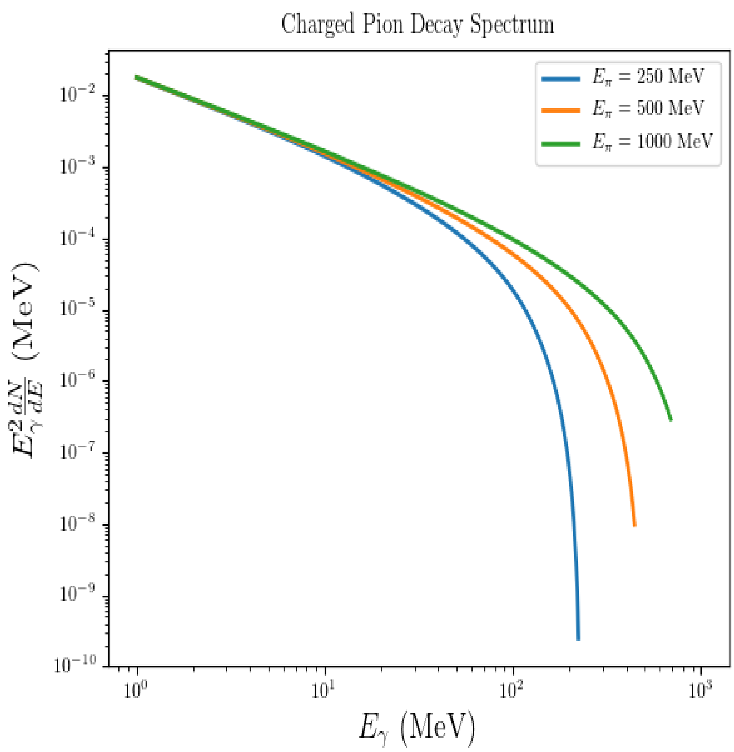
$$E_\gamma^{R\mu} = \gamma_1 E_\gamma^{R\pi}\left(1 - \beta_1 \cos\theta_\gamma^{R\pi}\right)$$

and

$$E_\gamma^{R\pi} = \gamma_2 E_\gamma^{L}\left(1 - \beta_2 \cos\theta_\gamma^{L}\right)$$

The limits on the photon energy are given by

$$0 \leq E_\gamma^L \leq \frac{m_\mu^2 - m_e^2}{2m_\mu}\gamma_1\gamma_2(1+\beta_1)(1+\beta_2)$$



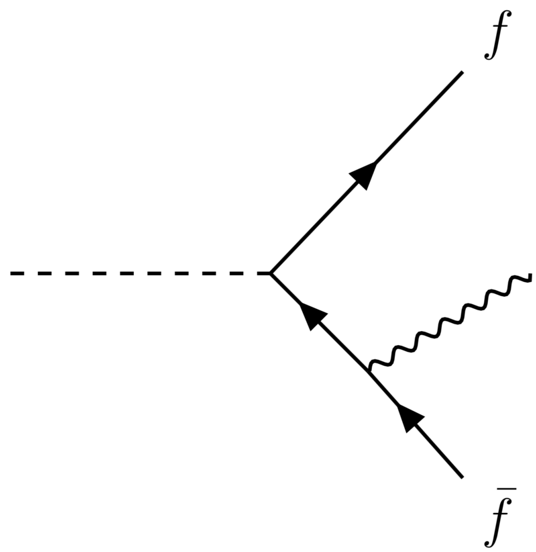
1.2.3 Neutral Pion

The dominant decay mode of the neutral pion is $\pi^0 \to \gamma\gamma$. In the laboratory frame, the spectrum is

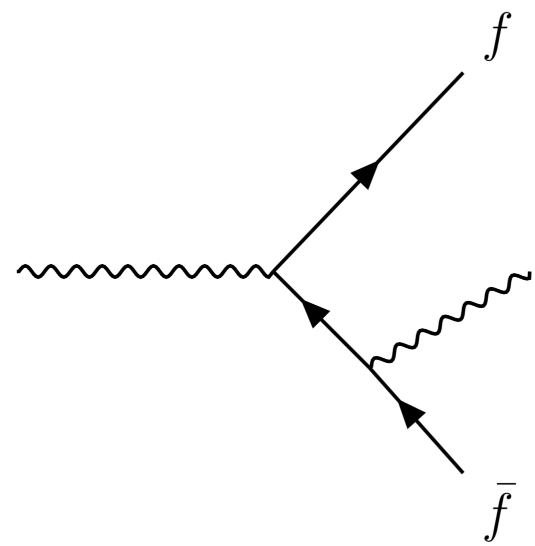
$$\frac{dN}{dE_\gamma} = \frac{2}{m_\pi \gamma\beta}$$

1.3 Final State Radiation

Along with computing decay spectra, hazma is able to compute final state radiation spectra from decays of off-shell mediators (scalar, psuedo-scalar, vector or axial-vector.) The relavent diagrams for such processes are

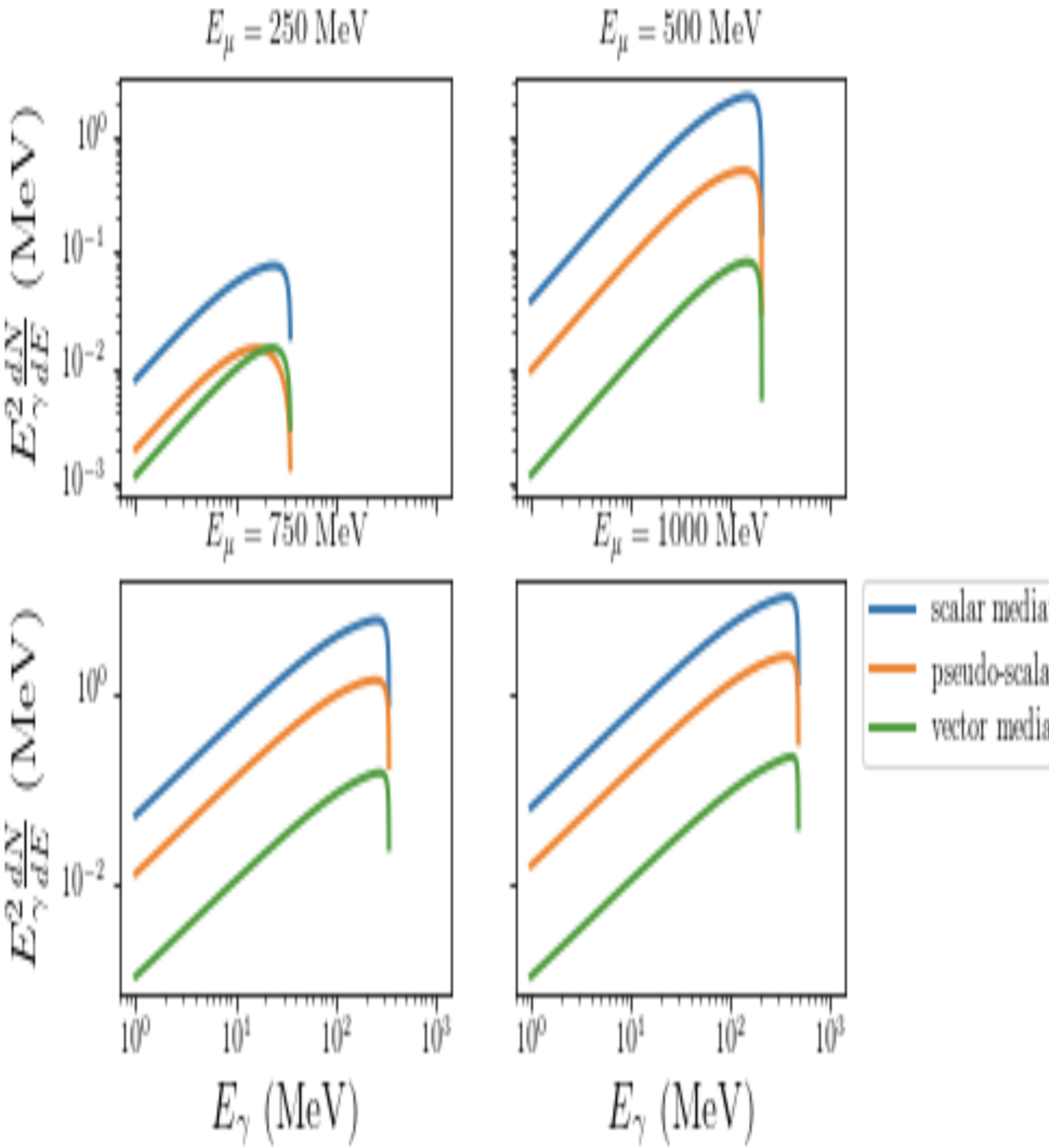


1. Scalar mediator



2. Vector mediator

Computing the matrix elements squared of these diagrams (including diagrams with the photon attached to the other fermion leg) and integrating over all variables except the photon energy yields $d\sigma(M^* \to \mu^+\mu^-\gamma)/dE$. To compute dN/dE, we divide $d\sigma(M^* \to \mu^+\mu^-\gamma)/dE$ by $\sigma(M^* \to \mu^+\mu^-)$.



1.3.1 References

CHAPTER

TWO

MODULES

The main modules of Hazma are the particle modules (electron, muon, charged pion, neutral pion, charged kaon and the neutral kaons) and the *gamma_ray* module. Each of the particle modules have two functions `decay_spectra` and `fsr` which produce the gamma ray spectra from radiative decays and final state radiation, respectively.

2.1 Gamma Ray (`hazma.gamma_ray`)

Module for computing gamma ray spectra from a many-particle final state.

- author : Logan Morrison and Adam Coogan
- date : December 2017

`hazma.gamma_ray.`**`gamma_ray`**(*particles*, *cme*, *eng_gams*, *mat_elem_sqrd=<function <lambda>>*, *num_ps_pts=1000*, *num_bins=25*)

Returns total gamma ray spectrum from a set of particles.

Blah Blah

Parameters **particles** : np.ndarray

List of particle names. Availible particles are 'muon', 'electron' 'charged_pion', 'neutral pion', 'charged_kaon', 'long_kaon', 'short_kaon'

cme [double] Center of mass energy of the final state in MeV.

eng_gams [np.ndarray[double, ndim=1]] List of gamma ray energies in MeV to evaluate spectra at.

mat_elem_sqrd [double(*func)(np.ndarray,)] Function for the matrix element squared of the proccess. Must be a function taking in a list of four momenta of size (num_fsp, 4). Default value is a flat matrix element.

num_ps_pts [int {1000}, optional] Number of phase space points to use.

num_bins [int {25}, optional] Number of bins to use.

Returns **spec** : np.ndarray

Total gamma ray spectrum from all final state particles.

rac{dN}{dE}(E_{gamma}) = sum_{i,j}P_{i}(E_{j})

rac{dN_i}{dE}(E_{gamma}, E_{j})

where i runs over the final state particles, j runs over energies sampled from probability distributions. $P_i(E_j)$ is the probability that particle i has energy E_j. The probabilities are computed using *hazma.phase_space_generator.rambo*. The total number of energies used is *num_bins*.

Examples

Example of generating a spectrum from a muon, charged kaon and long kaon with total energy of 5000 MeV.

```
>>> from hazma.gamma_ray import gamma_ray
>>> import numpy as np
>>>
>>> particles = np.array(['muon', 'charged_kaon', 'long_kaon'])
>>> cme = 5000.
>>> eng_gams = np.logspace(0., np.log10(cme), num=200, dtype=np.float64)
>>>
>>> spec = gamma_ray(particles, cme, eng_gams)
```

2.2 Muon (`hazma.muon`)

Module for computing gamma ray spectra from a muon.

@author - Logan Morrison and Adam Coogan @date - December 2017

`hazma.muon.`**`decay_spectra`**(*eng_gam*, *eng_mu*)

Compute dNdE from muon decay.

Compute dNdE from decay mu -> e nu nu gamma in the laborartory frame given a gamma ray engergy of `eng_gam` and muon energy of `eng_mu`.

Parameters **eng_gam** : numpy.ndarray

Gamma ray energy(ies) in laboratory frame.

eng_mu : double

Muon energy in laboratory frame.

Returns **spec** : numpy.ndarray

List of gamma ray spectrum values, dNdE, evaluated at `eng_gam` given muon energy `eng_mu`.

Examples

Calculate spectrum for single gamma ray energy

```
>>> from hazma import muon
>>> eng_gam, eng_mu = 200., 1000.
>>> spec = muon.decay_spectra(eng_gam, eng_mu)
```

Calculate spectrum for array of gamma ray energies

```
>>> from hazma import muon
>>> import numpy as np
>>> eng_gams = np.logspace(0.0, 3.0, num=200, dtype=float)
```

```
>>> eng_mu = 1000.
>>> spec = muon.decay_spectra(eng_gams, eng_mu)
```

`hazma.muon.`**`fsr`**`(`*eng_gam*, *cme*, *mediator='scalar'*`)`

Compute muon fsr spectrum.

Compute final state radiation spectrum dN/dE from decay of an off-shell mediator (scalar, psuedo-scalar, vector or axial-vector) into a pair of muons.

Paramaters eng_gam (float or np.ndarray) : Gamma ray energy(ies) in laboratory frame. cme (float) : Center of mass energy or mass of the off-shell mediator. mediator (string) : Mediator type : scalar, psuedo-scalar, vector or axial-vector.

Returns spec (np.ndarray) : List of gamma ray spectrum values, dNdE, evaluated at *eng_gams* given a center of mass energy *cme*.

Examples dNdE for a single gamma ray energy from scalar mediator.

```
>>> from hazma import muon
>>> eng_gam, cme = 200., 1000.
>>> spec = muon.fsr(eng_gam, cme, 'scalar')
```

dNdE for list of gamma ray energies from vector mediator.

```
>>> from hazma import muon
>>> eng_gams = np.logspace(0.0, 3.0, num=1000, dtype=float)
>>> cme = 1000.
>>> spec = muon.fsr(eng_gams, cme, 'scalar')
```

2.3 Electron (`hazma.electron`)

Module for computing gamma ray spectra from an electron.

@author - Logan Morrison and Adam Coogan @date - December 2017

`hazma.electron.`**`decay_spectra`**`(`*eng_gam*, *eng_mu*`)`

Returns zero. Electron is stable.

`hazma.electron.`**`fsr`**`(`*eng_gam*, *cme*, *mediator='scalar'*`)`

Compute electron fsr spectrum.

Compute final state radiation spectrum dN/dE from decay of an off-shell mediator (scalar, psuedo-scalar, vector or axial-vector) into a pair of electrons.

Returns **spec** : (np.ndarray)

List of gamma ray spectrum values, dNdE, evaluated at *eng_gams* given a center of mass energy *cme*.

Examples

dNdE for a single gamma ray energy from scalar mediator.

```
>>> from hazma import electron
>>> eng_gam, cme = 200., 1000.
>>> spec = electron.fsr(eng_gam, cme, 'scalar')
```

dNdE for list of gamma ray energies from vector mediator.

```
>>> from hazma import electron
>>> eng_gams = np.logspace(0.0, 3.0, num=1000, dtype=float)
>>> cme = 1000.
>>> spec = electron.fsr(eng_gams, cme, 'scalar')
```

2.4 Charged Pion (`hazma.charged_pion`)

Module for computing gamma ray spectra from a charged pion.

@author - Logan Morrison and Adam Coogan @date - December 2017

`hazma.charged_pion.`**`decay_spectra`**`(`*eng_gam*, *eng_pi*`)`

Compute dNdE from charged pion decay.

Compute dNdE from decay pi -> mu nu -> e nu nu g in the laborartory frame given a gamma ray engergy of *eng_gam* and muon energy of *eng_pi*.

Returns spec : double np.ndarray

List of gamma ray spectrum values, dNdE, evaluated at *eng_gams* given charged pion energy *eng_pi*.

Examples

Calculate spectrum for single gamma ray energy

```
>>> from hazma import charged_pion
>>> eng_gam, eng_pi = 200., 1000.
>>> spec = charged_pion.decay_spectra(eng_gam, eng_pi)
```

Calculate spectrum for array of gamma ray energies

```
>>> from hazma import charged_pion
>>> import numpy as np
>>> eng_gams = np.logspace(0.0, 3.0, num=200, dtype=float)
>>> eng_pi = 1000.
>>> spec = charged_pion.decay_spectra(eng_gams, eng_pi)
```

`hazma.charged_pion.`**`fsr`**`(`*eng_gam*, *cme*, *mediator='scalar'*`)`

NOT YET IMPLEMENTED!

2.5 Neutral Pion (`hazma.neutral_pion`)

Module for computing gamma ray spectra from a neutral pion.

@author - Logan Morrison and Adam Coogan @date - December 2017

`hazma.neutral_pion.`**`decay_spectra`**`(`*eng_gam*, *eng_pi*`)`

Compute dNdE from neutral pion decay.

Compute dNdE from decay pi0 -> gamma gamma in the laborartory frame given a gamma ray engergy of *eng_gam* and neutral pion energy of *eng_pi*.

Returns spec : np.ndarray

List of gamma ray spectrum values, dNdE, evaluated at *eng_gams* given neutral pion energy *eng_pi*.

`hazma.neutral_pion.fsr(eng_gam, cme, mediator='scalar')`
Returns zero.

2.6 Charged Kaon (`hazma.charged_kaon`)

Module for computing gamma ray spectra from a charged kaon.

- author : Logan Morrison and Adam Coogan
- date : December 2017

`hazma.charged_kaon.decay_spectra(eng_gam, eng_k)`
Compute dNdE from charged kaon decay.

Compute dNdE from decay of charged kaon through K -> X in the laboratory frame given a gamma ray engergy of *eng_gam* and charged kaon energy of *eng_k*. The decay modes impemendted are * k -> mu + nu * k -> pi + pi0 * k -> pi + pi + pi * k -> pi0 + e + nu * k -> pi0 + mu + nu * k -> pi + pi0 + pi0

Returns spec : np.ndarray

List of gamma ray spectrum values, dNdE, evaluated at *eng_gams* given muon energy *eng_mu*.

`hazma.charged_kaon.fsr(eng_gam, cme, mediator='scalar')`
NOT YET IMPLEMENTED!

2.7 Long Kaon (`hazma.long_kaon`)

Module for computing gamma ray spectra from a charged kaon.

- author : Logan Morrison and Adam Coogan
- date : December 2017

`hazma.charged_kaon.decay_spectra(eng_gam, eng_k)`
Compute dNdE from charged kaon decay.

Compute dNdE from decay of charged kaon through K -> X in the laboratory frame given a gamma ray engergy of *eng_gam* and charged kaon energy of *eng_k*. The decay modes impemendted are * k -> mu + nu * k -> pi + pi0 * k -> pi + pi + pi * k -> pi0 + e + nu * k -> pi0 + mu + nu * k -> pi + pi0 + pi0

Returns spec : np.ndarray

List of gamma ray spectrum values, dNdE, evaluated at *eng_gams* given muon energy *eng_mu*.

`hazma.charged_kaon.fsr(eng_gam, cme, mediator='scalar')`
NOT YET IMPLEMENTED!

2.8 Short Kaon (`hazma.short_kaon`)

Module for computing gamma ray spectra from a charged kaon.

- author : Logan Morrison and Adam Coogan
- date : December 2017

`hazma.charged_kaon.`**`decay_spectra`**(*eng_gam*, *eng_k*)

Compute dNdE from charged kaon decay.

Compute dNdE from decay of charged kaon through K -> X in the laboratory frame given a gamma ray engergy of *eng_gam* and charged kaon energy of *eng_k*. The decay modes impemendted are * k -> mu + nu * k -> pi + pi0 * k -> pi + pi + pi * k -> pi0 + e + nu * k -> pi0 + mu + nu * k -> pi + pi0 + pi0

Returns **spec** : np.ndarray

List of gamma ray spectrum values, dNdE, evaluated at *eng_gams* given muon energy *eng_mu*.

`hazma.charged_kaon.`**`fsr`**(*eng_gam*, *cme*, *mediator='scalar'*)

NOT YET IMPLEMENTED!

CHAPTER

THREE

USAGE

Will update soon!

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

h

hazma.charged_kaon, 13
hazma.charged_pion, 12
hazma.electron, 11
hazma.gamma_ray, 9
hazma.muon, 10
hazma.neutral_pion, 12

INDEX

D

decay_spectra() (in module hazma.charged_kaon), 13, 14
decay_spectra() (in module hazma.charged_pion), 12
decay_spectra() (in module hazma.electron), 11
decay_spectra() (in module hazma.muon), 10
decay_spectra() (in module hazma.neutral_pion), 12

F

fsr() (in module hazma.charged_kaon), 13, 14
fsr() (in module hazma.charged_pion), 12
fsr() (in module hazma.electron), 11
fsr() (in module hazma.muon), 11
fsr() (in module hazma.neutral_pion), 13

G

gamma_ray() (in module hazma.gamma_ray), 9

H

hazma.charged_kaon (module), 13
hazma.charged_pion (module), 12
hazma.electron (module), 11
hazma.gamma_ray (module), 9
hazma.muon (module), 10
hazma.neutral_pion (module), 12