Problem Set 2 problems

Question 1: Relativity and Particle Decays

Learning objectives

In this question you will:

- Review relativistic expressions relevant for determining the mass and lifetime of a particle from its decay products
- gain experience in using python to analyze data provided in a text file

1a.

The decays of particles with lifetimes longer than ~ 0.5 ps can be observed in high resolution particle detectors. By measuring many particle decays, properties such as the decaying particle's mass and lifetime can be determined.

The file decayData.dat contains a set of simulated observations of particle decays that come from one specific species of hadron, which we designate particle X. The X is observed through its decay $X \rightarrow p\pi^-$ where the p is a proton. All the X particles are produced at the origin $(x = 0, y = 0, z = 0)$ but they have with a range of momenta. The position of the decay and the momentum of the proton and π^- are measured.

The following code reads this data file and puts the data into a form that can be easily used in python:

In [1]:
```
import math
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

# Parse the input file.
file = "decayData.dat"

#Each row corresponds to one event. The columns are:
# x-position of the decay vertex in cm
# y-position of the decay vertex in cm
# z-position of the decay vertex in cm
# species of first particle (always a proton)
# p1x: x-momentum of the proton produced in the decay in GeV
# p1y: y-momentum of the proton produced in the decay in GeV
# p1z: z-momentum of the proton produced in the decay in GeV
# species of second particle (always a pi^-)
# p2x: x-momentum of the pi^- produced in the decay in GeV
# p2x: y-momentum of the pi^- produced in the decay in GeV
# p2x: z-momentum of the pi^- produced in the decay in GeV


inMeta = False
vx = []
vy = []
vz = []
p1x = []
p1y = []
p1z = []
p2x = []
p2y = []
p2z = []


inMeta = True
for line in open(file,"r"):
    line = line.strip()
    info = line.split(",")
    if inMeta and ("<metadata>" in info[0]):
        inMeta = True
    elif inMeta and ("</metadata>" in info[0]):
        inMeta = False
    elif not inMeta:
        vx.append(float(info[0]))
        vy.append(float(info[1]))
        vz.append(float(info[2]))
        p1x.append(float(info[4]))
        p1y.append(float(info[5]))
        p1z.append(float(info[6]))
        p2x.append(float(info[8]))
        p2y.append(float(info[9]))
        p2z.append(float(info[10]))
massPiInGeV = 0.13957
massProtonInGeV = 0.93827
```

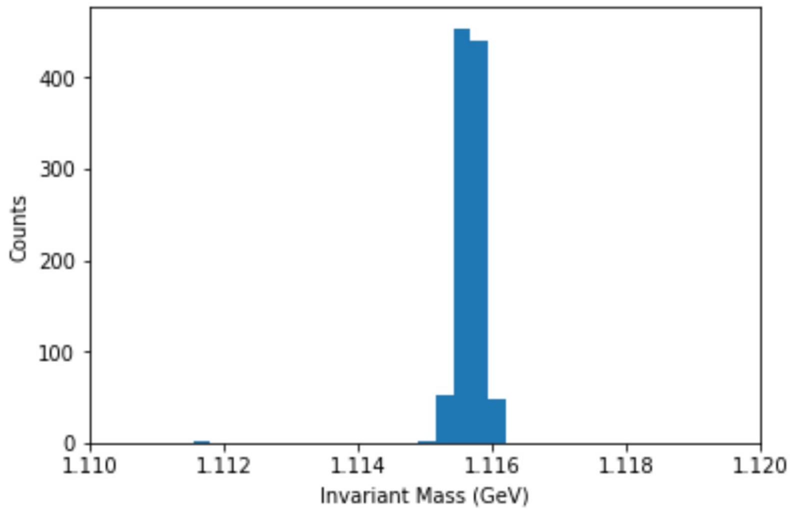
Verify that all these events correspond to the decay of a particle of a specific species by making a histogram of the invariant mass of the decays.

In [2]:
```
# '1' = proton, '2' = pion
```

In [3]:
```
# cm --> meters
vx = np.array(vx)/100
vy = np.array(vy)/100
vz = np.array(vz)/100
p1x = np.array(p1x)
p1y = np.array(p1y)
p1z = np.array(p1z)
p2x = np.array(p2x)
p2y = np.array(p2y)
p2z = np.array(p2z)
```

In [4]:
```
# invariant mass formula
# E^2 - |pvec|^2 = m^2
proton_energies = np.sqrt(massProtonInGeV**2 + p1x**2 + p1y**2 + p1z**
2)
pion_energies = np.sqrt(massPiInGeV**2 + p2x**2 + p2y**2 + p2z**2)

X_energies = proton_energies + pion_energies
Xpx = p1x + p2x
Xpy = p1y + p2y
Xpz = p1z + p2z
Xpmag = np.sqrt(Xpx**2 + Xpy**2 + Xpz**2)

X_invmass = np.sqrt(X_energies**2 - Xpmag**2)
```

In [5]:
```
plt.figure()
plt.hist(X_invmass, bins = 100)
plt.xlabel("Invariant Mass (GeV)")
plt.xlim(1.110, 1.12)
plt.ylabel("Counts")
plt.show()
print('Mean invariant mass =', np.mean(X_invmass), 'GeV')
```




```
Mean invariant mass = 1.1156460781296815 GeV
```

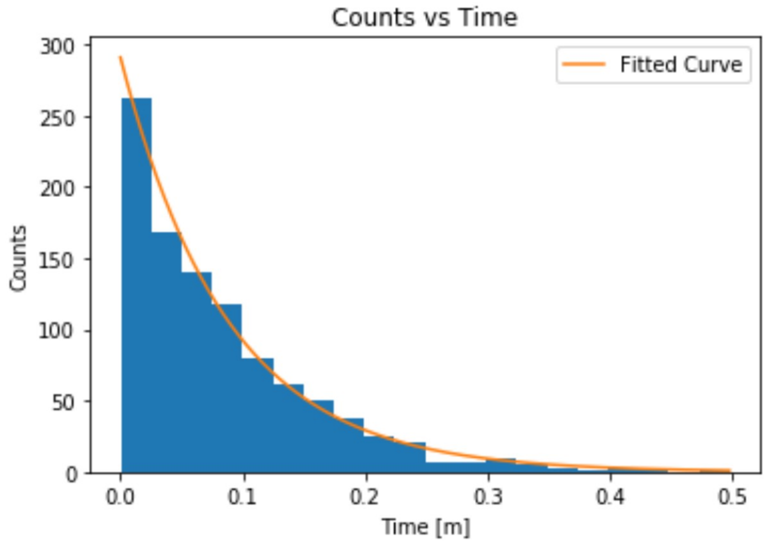
1b.

Using these data, determine the lifetime of the X particle. What evidence do you have that the X has a decay distribution consistent with a single species with one lifetime? Note: your answer to this part does not have to be very detailed. A simple graph and a sentence or two of explanation is sufficient

In [242]:
```
from scipy.optimize import curve_fit
```

In [268]:
```
hbar = 6.582*10**(-16) *10**(-9)   # GeV*s
c = 3*(10**8) # m/s
```

In [278]:
```
# v = p/m
# position magnitude/tau = v = p/m  <--- this works because all X's st
arted at the origin of (0,0,0)
# tau = lifetime = position (magnitude) * mass / momentum
distances = np.sqrt(vx**2 + vy**2 + vz**2) # m
#NUdistances = (distances * 10**9)/(hbar * c) # GeV^-1
times = distances * X_invmass/ Xpmag # m

# we expect the decay to fit to an exponential decay
def decay(times, Norm, tau):
    return (Norm/tau)*np.exp(-times/tau)

num_bins = 20


plt.figure()
plt.title("Counts vs Time")

n, bins, patches = plt.hist(times,num_bins)#, density = True)
bin_centers = 0.5*(bins[1:] + bins[:-1])

#plt.hist(tau, bins = 20, density = True)
plt.xlabel('Time [m]')
plt.ylabel("Counts")

fittedparams, hessian = curve_fit(decay, bin_centers, n)
param_err = np.sqrt(np.diag(hessian))

t = np.linspace(0, np.max(times), 1000)
plt.plot(t, decay(t, fittedparams[0], fittedparams[1]), label='Fitted
Curve')
plt.legend()
plt.show()

print("Normalization factor = ", fittedparams[0])
print('Mean Lifetime (tau) = ', fittedparams[1], 'm')
#print('in natural units: tau = ', fittedparams[1]/hbar, 'GeV^-1')
```




```
Normalization factor =  25.30662647550137
Mean Lifetime (tau) =  0.08700914342754491 m
```

Evidence: The data fits well to a single exponential decay curve; thus, it's highly likely that the decay rate of X corresponds to that of 1 species.

Question 2: π^0 Decay

Learning objectives

In this question you will:

- Review basic concepts of Special Relativity and Lorentz Boosts
- Apply these concepts to the case of $\pi^0 \to \gamma\gamma$ decay
- Learn techniques needed to simulate the decay of an ensemble of π^0s with non-zero momentum

Adapted from Perkins 4^{th} Edition Problem 1.4

In this problem you will derive an expression for the distribution of photon energies produced in the decays of π^0s that are moving with fixed momentum. Then, you will learn how to create a simulated sample of such π^0 decays. Note: In this problem, we will use natural units where $\hbar = c = 1$.

2a.

A particle beam consists of π^0's all with energy E_{lab} and all traveling in the $+z$ direction. Find an expression for the energy of the photons produced from the π^0 decays as a function of m_π, E_{lab} and θ^* (the angle of emission of the photon with respect to the z-axis in the pion rest frame). Using this expression, show that the lab energy spectrum of the photons is flat, extending from $E_{lab}\left(1+\beta\right)/2$ to $E_{lab}\left(1-\beta\right)/2$, where β is the velocity of the π^0 in the lab frame.

In the CM frame:

$$\overrightarrow{k_1} = -\overrightarrow{k_2}$$

This is because the resulting photons have zero mass, but the momentum must be conserved in the CM frame

$$m_{\pi^0} = k_1^0 + k_2^0 = 2k_1^0$$

where $k_1 = (k_0^1; \overrightarrow{k_1})$. Thus,

$$k_1^0 = \frac{m_\pi^0}{2}$$
$$\therefore k_1 = \frac{m_{\pi^0}}{2}(1, \vec{n})$$

Here, $\vec{n}$ the vector with direction θ^* from the z-axis. k_2 has a similar expression but pointing in the opposite direction, as to (again) conserve momentum.

Now, we have to boost into the lab frame to get the observed energy spectrum (for "photon 1"):

$$E_{lab_1} = \gamma(k_1^0 + \vec{\beta}\cdot\overrightarrow{k_1})$$

Plugging in the four momentum k_1 from above, we have:

$$E_{lab_1} = \frac{m_{\pi^0}\gamma}{2}(1+\beta cos(\theta^*))$$

as the observed energy spectrum of γ_1. From here, we recognize that if we Lorentz boost the original pion energy, we have

$$E_{lab_\pi} = \gamma m_{\pi^0}$$

Thus, we can plug it into the photon's energy equation

$$E_{lab_1} = \frac{m_{\pi^0}\gamma}{2}(1+\beta cos(\theta^*)) = \frac{E_{lab_\pi}}{2}(1+\beta cos(\theta^*))$$

which has the range we want. Applying the same transformation for γ_2's energy, we have almost the same expression (difference only by $-\vec{n}$ instead of $\vec{n}$):

$$E_{lab_2} = \frac{m_{\pi^0}\gamma}{2}(1-\beta cos(\theta^*)) = \frac{E_{lab_\pi}}{2}(1-\beta cos(\theta^*))$$

2b.

Find an expression for the disparity D (the ratio of the energy of the higher energy photon to the energy of the lower energy) and show that in the relativistic limit $\beta \approx 1$, $D > 3$ in half the decays and $D > 7$ in one quarter

The maximum photon energy possible should be $constant \times (1+\beta)$ and the minimum $constant \times (1-\beta)$ (where the constant is just the term out front from the above E_{lab_1} expression. So the higher energy one, for the same θ^* should be γ_1. Thus, the disparity D is:

$$D = \frac{E_{higher}}{E_{lower}} = \frac{1+\beta cos(\theta^*)}{1-\beta cos(\theta^*)}$$

Assumming $\beta \approx 1$, then

$$D = \frac{1+cos(\theta^*)}{1-cos(\theta^*)}$$

To find disparity after a certain amount of decays, we need to know its distribution, which means we need to know $\frac{dN}{dD}$. We can do this by figuring out the result of

$$\frac{dN}{dD} = \frac{dN}{dcos(\theta^*)}\frac{dcos(\theta^*)}{dD}$$

This can be simplified (visually) by setting a new variable $x = \cos(\theta^*)$ So

$$D = \frac{1+x}{1-x}$$

$$\frac{dD}{dx} = \frac{2}{(1-x)^2}$$

$$\frac{dN}{dD} = \frac{dN}{dx}(\frac{dD}{dx})^{-1} = \frac{dN}{dx}\frac{(1-x)^2}{2}$$

Rewriting x to be in terms of D

$$x = \frac{D-1}{D+1}$$

so

$$\frac{dN}{dD} = \frac{dN}{dx}(\frac{dD}{dx})^{-1} = \frac{dN}{dx}\frac{2}{(1+D)^2}$$

Finally, we can integrate to get N out:

$$N(D > D_0) = \int \frac{dN}{dx}\frac{2}{(1+D)^2}dD = \frac{dN}{dx}\frac{2}{1+D_0}$$

We integrate it like this because we want to know the rest of the distribution ABOVE a certain D value, and also we're integrating to infinity.

So, we're given $D > 3$ and $D > 7$, so:

$$N(D > 3) = \frac{dN}{dx}\frac{2}{1+3} = \frac{1}{2}\frac{dN}{dx}$$

$$N(D > 7) = \frac{dN}{dx}\frac{2}{1+7} = \frac{1}{4}\frac{dN}{dx}$$

We see that the result checks out with the initial claim "$D > 3$ in half the decays and $D > 7$ in one quarter"

2c.

It is often useful for physicists to simulate experimental data. Such simulations allow us generate an ensemble of events corresponding to a given physical process and to study them. Generated events can be passed through a simulated detector that has imperfections (finite resolution, missing channels, incomplete angular coverage, etc) and the effect of such imperfections on our measurements can be assessed. This problem is our first example of creating such simulated data. Our simulation will be quite simple but the concepts developed here will be used through the semester.

Assume we have a beam of 10000 π^0 all with energy 5 GeV. Simulate the decay of these pions and plot (histogram) the following distributions:

- The energies of the photons produced in the π^0 decay
- The disparity of the decays
- The angles θ between the momenta of the photons and that of the π^0 in the lab frame.

Hints:

- For each decay, first simulate the decay in the pion center of mass and then Lorentz boost to the lab frame
- Since in the rest frame of the pion, the decay is isotropic, the distribution of $\cos\theta^*$ is uniformly distributed. If for each event you pull a random number uniformly distributed between 0 and 1 and set $\cos\theta^*$ for that event equal to the random number, the decays will have the right distribution.
- In principle, you could find the ϕ^* angle for each decay by pulling a second uniformly distributed random number, but for this problem you will not need the x and y components of the photon momentum separately so you don't need to do this.

From 2a, we have

$$E_{lab_\pi} = m_\pi \gamma \to \gamma = \frac{E_{lab_\pi}}{m_\pi}$$

Thus, we have

In [148]:
```
Elab = 5 #GeV
m_pi = 0.135 #GeV
gamma = Elab/m_pi # GeV/GeV, so unitless
print('Gamma =', gamma)
beta = np.sqrt(1 - (1/gamma**2))
print('Beta =', beta)
```

```
Gamma = 37.03703703703704
Beta = 0.9996354335456502
```

In [154]:
```
np.max(np.random.rand(N))
```

Out[154]:
```
0.9997901026780452
```

In [147]:
```
beta
```
Out[147]:
```
0.9996354335456502
```

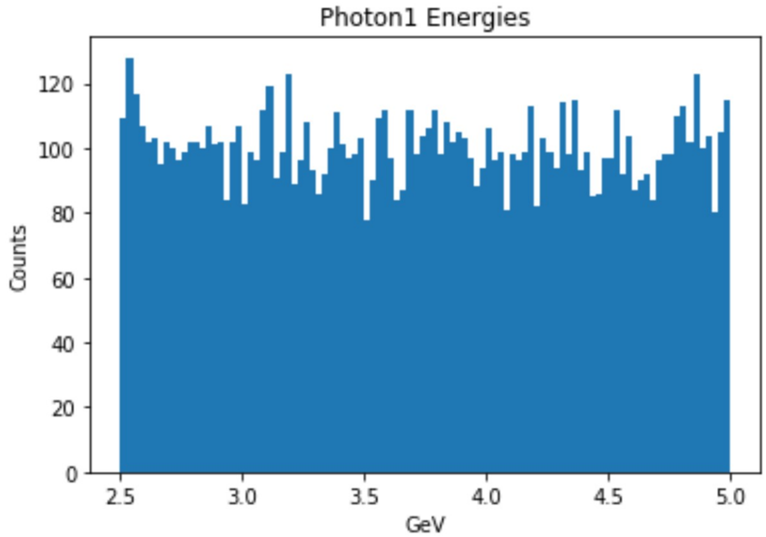
In [294]:
```
# input N pions, output N photon1 energies and N photon2 energies
def photon_energies(N, m, E):
    gamma = E/m
    beta = np.sqrt(1 - (1/gamma**2))

    # beta is very close to 1; the thing is, the function will break l
ater on due to the angle transformation
    # this is because it involves np.arccos(angle that is very very sl
ightly larger than 1), and that function will
    # throw an exception and break. Thus, the distribution is still un
iform but will disclude a very small portion of
    # the top end towards 1 (< 10^-4 off the top)
    costheta_star = np.random.uniform(0, beta, N)

    E1 = (m*gamma/2)*(1 + beta*costheta_star)
    E2 = (m*gamma/2)*(1 - beta*costheta_star)
    D = E1/E2
    # angle in lab frame, which needs another Lorentz boost

    angles = np.arccos((beta + costheta_star)/(2*beta))
    open_angle = np.arccos((beta + costheta_star)/(1+costheta_star))
#     for i in range(len(costheta_star)):
#         if (beta + costheta_star[i])/(2*beta) > 1:
#             print((beta + costheta_star[i])/(2*beta))
#             print('warning')
#         angles.append(np.arccos((beta + costheta_star[i])/(2*beta)))

#     angles = np.array(angles)

    return E1, E2, D, angles, open_angle
```

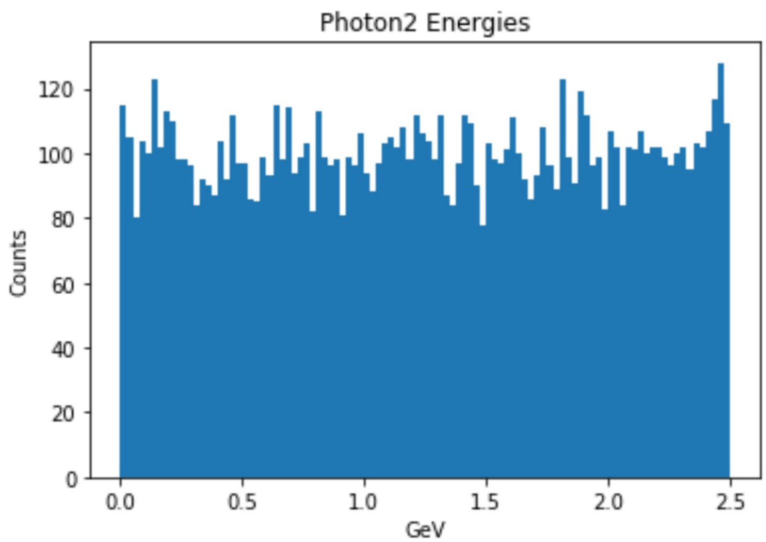
In [1]:
```
N = 10000
p1energies, p2energies, disp, angles, open_lol = photon_energies(N = N,
m = m_pi, E = Elab)
```

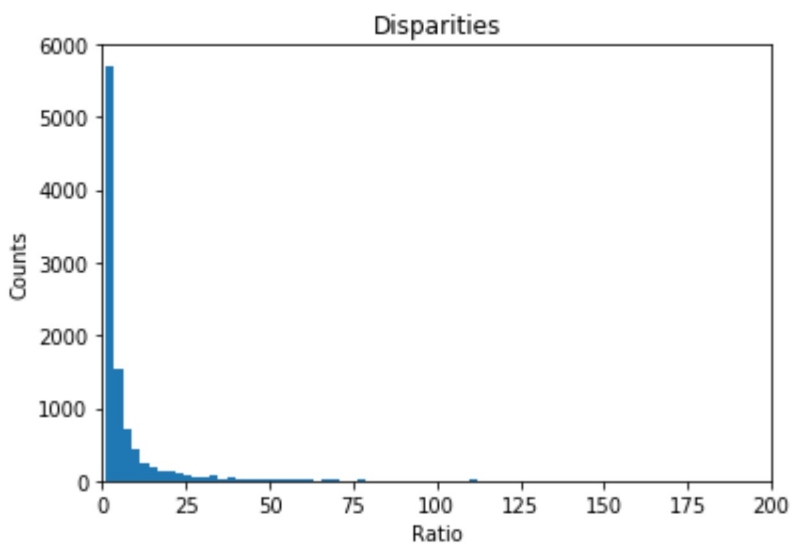
```
----------------------------------------------------------------------
------
NameError                                 Traceback (most recent call
last)
<ipython-input-1-95351326a084> in <module>
      1 N = 10000
----> 2 p1energies, p2energies, disp, angles, open_lol = photon_energ
ies(N = N, m = m_pi, E = Elab)

NameError: name 'photon_energies' is not defined
```

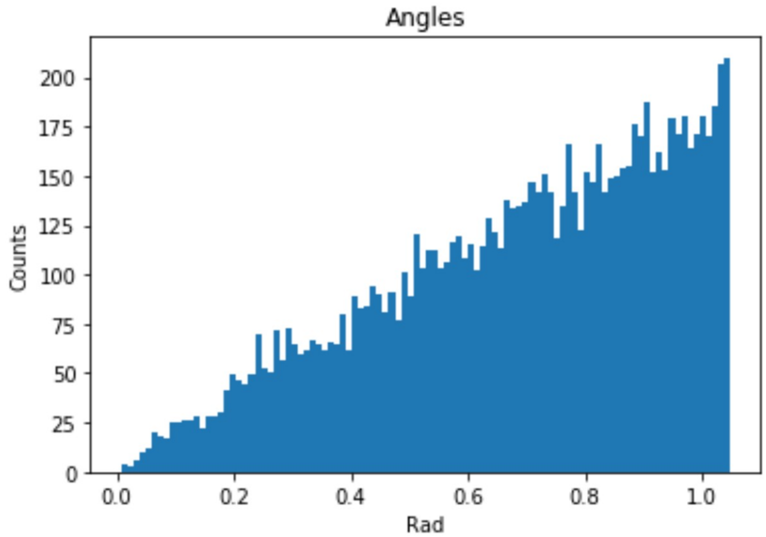
In [296]:
```
np.max(disp)
```
Out[296]:
```
2591.6791358147
```

In [297]:
```
bins = int(N/100)

plt.figure()
plt.title('Photon1 Energies')
plt.hist(p1energies, bins)
plt.xlabel('GeV')
plt.ylabel('Counts')

plt.figure()
plt.title('Photon2 Energies')
plt.hist(p2energies, bins)
plt.xlabel('GeV')
plt.ylabel('Counts')

plt.figure()
plt.title('Disparities')
plt.hist(disp, bins*10)
plt.xlabel('Ratio')
plt.xlim(0,200)
plt.ylabel('Counts')

plt.figure()
plt.title('Angles')
plt.hist(angles, bins)
plt.xlabel('Rad')
plt.ylabel('Counts')


plt.show()
```


Photon1 Energies
Counts
GeV
0
20
40
60
80
100
120
2.5
3.0
3.5
4.0
4.5
5.0


Photon2 Energies
Counts
GeV
0
20
40
60
80
100
120
0.0
0.5
1.0
1.5
2.0
2.5


Disparities
Counts
Ratio
0
1000
2000
3000
4000
5000
6000
0
25
50
75
100
125
150
175
200



Note: For the Disparities, the maximum value is

In [298]:
```
dispmax = np.max(disp)
print(dispmax)
```

```
2591.6791358147
```

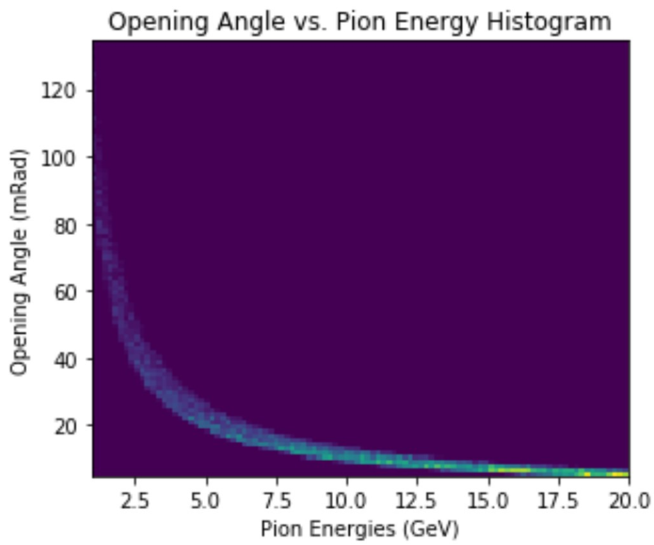
Which is much larger than 200, but if xlim is set to dispmax, then the histogram doesn't look like anything other than a single bar at 1.

2d.

Modify your simulation so that instead of having a fixed energy beam, the π^0 energy is uniformly distributed between 1 and 20 GeV. Make a 2D histogram of the opening angle between the two photons (measured in milli-radians) as a function of the π^0 energy.

In [299]:
```
energydist = np.random.uniform(1,20,N) #GeV
```

In [300]:
```
# 2D histogram
distp1, distp2, distdisp, distangles, open_angles = photon_energies(N
= N, m = m_pi, E = energydist)
distangles *= 1000 #rad to millirad
open_angles *= 1000 #rad to millirad

plt.figure()
plt.hist2d(energydist, open_angles, bins = 100)
plt.xlabel("Pion Energies (GeV)")
plt.ylabel('Opening Angle (mRad)')
plt.title("Opening Angle vs. Pion Energy Histogram")
plt.colorbar()
plt.show()
```

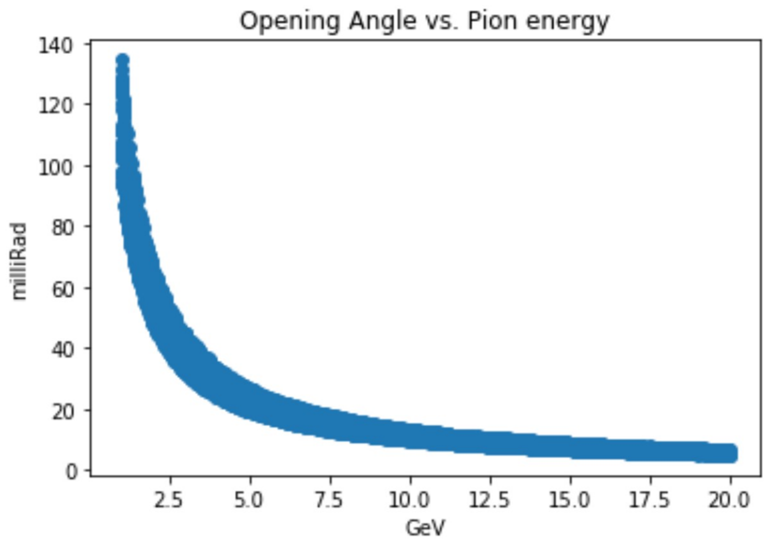


2e.

In the ATLAS detector, photons are identifed in the electromagnetic calorimeter by looking for a narrow energy cluster. Assume that two photons will be *merged* into a single cluster if their opening angles differ by less than 75 milli-radians. Using your scatter plot above, estimate the maximum energy π^0 for which the decay photons can be cleanly separated. (Note: the ATLAS detector is more complicated than the description presented in this problem, having different granularities in θ and ϕ directions. Moreover, the experiment can *identify* π^0 at higher energies than suggested here by looking at the width of the merged energy deposit from the two clusters.)

Given that we weren't asked to make a scatter plot from above, here is one below:

In [302]:
```
plt.figure()
plt.scatter(energydist, open_angles)
plt.title("Opening Angle vs. Pion energy")
plt.xlabel("GeV")
plt.ylabel("milliRad")
plt.show()
```



In [314]:
```
from scipy.interpolate import interp1d
```

In [303]:
```
# just showing that minimum, which should be less than 75mRad; i.e. at
least 1 "merge" occurs
np.min(distangles)
```

Out[303]: 13.5509648884437

In [316]:
```
for i in range(len(open_angles)):
    if i == 75:
        print(energydist[i])

print(interp1d(open_angles, energydist)(75))
print('Minimum energy: ', interp1d(open_angles, energydist)(75), 'GeV
')
```

```
16.551317559573146
1.4655111176788576
Minimum energy:  1.4655111176788576 GeV
```

Question 3: Mandelstam Variables

Learning objectives

In this question you will:

- Review the definitions of the Mandelstam Variables
- Apply relativistic formulae to derive and important relationship between these variables

In the two-to-tow process $1+2 \rightarrow 3+4$ the Mandelstan variable are define:

$$s = (p_1 + p_2)^2$$
$$t = (p_1 - p_3)^2$$
$$u = (p_1 - p_4)^2$$

3a.

Show that s is the square of the center-or-mass energy of the system $1+2$

Center of mass frame dictates that $\overrightarrow{p_1} = -\overrightarrow{p_2}$ (i.e. 3 momenta cancel out)
Regular vector addition first, then split into the scalar portion and the 3-vector portion.
Apply squaring last. Result:

$$s = (p_1 + p_2)^2 = (E_1 + E_2)^2 + (\overrightarrow{p_1} + \overrightarrow{p_2})^2$$

Plug in CM frame constraint:

$$s = (p_1 + p_2)^2 = (E_1 + E_2)^2 + (\overrightarrow{p_1} - \overrightarrow{p_1})^2$$
$$\therefore s = (E_1 + E_2)^2$$

3b.

Show that

$$s + t + u = m_1^2 + m_2^2 + m_3^2 + m_4^2$$

$$s+t+u=(p_1+p_2)^2+(p_1-p_3)^2+(p_1-p_4)^2$$
$$s+t+u=p_1^2+p_2^2+2p_1\cdot p_2+p_1^2+p_3^2-2p_1\cdot p_3+p_1^2+p_4^2-2p_1\cdot p_2$$

Square of the four momenta = square of the mass:

$$p_i^2=m_i^2$$

And 4 momentum conservation:

$$p_1+p_2=p_3+p_4$$

Thus,

$$s+t+u=3m_1^2+m_2^2+m_3^2+m_4^2+2p_1\cdot p_2-2p_1\cdot p_3-2p_1\cdot p_4$$
$$s+t+u=3m_1^2+m_2^2+m_3^2+m_4^2+2p_1\cdot(p_2-p_3-p_4)$$
$$s+t+u=3m_1^2+m_2^2+m_3^2+m_4^2+2p_1\cdot(-p_1)$$
$$s+t+u=3m_1^2+m_2^2+m_3^2+m_4^2-2m_1^2$$
$$\therefore s+t+u=m_1^2+m_2^2+m_3^2+m_4^2$$

Question 4: β-decay and the uncertainty principle

Learning objectives

In this question you will:

- Apply the uncertainty principle to the β-decay process to make an important conclusion

In the period before the discover of the neutron, many people thought that the nucleus consisted of protons and {\it electrons}, with the atomic number equal to the excess number of protons over electrons. This view seemed to be supported by the observation that in nuclear β-decay electrons are emitted and the charge of the nucleus changes so that overall charge is conserved. Use the position-momentum uncertainty principle, $\Delta x\Delta p\geq\hbar$, to estimate the minimum momentum of an electron confined to a nucleus (radius 10^{-13} cm). From the energy-momentum relation (in natural units) $E^2=p^2+m^2$ determine the corresponding energy and compare it with that emitted in, say, the β-decay of tritium. This result convinced some people (correctly) that the beta-decay electron could {\it not} have been rattling around inside the nucleus, but must be produced in the disintegration itself. Note: You can find a plot of the β-decay spectrum of tritium in Figure 1 of https://cerncourier.com/a/a-voyage-to-the-heart-of-the-neutrino/ (https://cerncourier.com/a/a-voyage-to-the-heart-of-the-neutrino/) That article also provides a nice introduction to the KATRIN experiment. KATRIN is designed to measure or set stringent limits on the mass of the ν_e

$$\Delta p\geq\frac{\hbar}{\Delta x}$$

Max Δx minimizes Δp. The maximum allowed Δx in that radius is

$$\Delta x=2\cdot 10^{-13}cm=2\cdot 10^{-15}m$$

In [258]:
```
m_e
```

Out[258]: 0.000511

In [276]:
```
hbar = 6.582*10**-16 # eV *s
x = 2*10**-15 # m
c = 3*10**8 #m/s
p = hbar/x # eV*s/m
print('minimum momentum change:', p, 'eV m/s')
pc = p*c # eV
m_e = 511*10**-6 # eV/c^2
m_ecc = m_e*(1**2) # eV/c^2 times two "units of c" = eV

E = np.sqrt(pc**2 + m_ecc**2) #eV
print('Energy of electron:', E*10**-9, 'GeV')
```

```
minimum momentum change: 0.32909999999999995 eV m/s
Energy of electron: 0.09872999999999998 GeV
```

β-decay of tritium is on the order of KeV. The energy of an electron trapped in that nucleic radius is ~0.1GeV, which is many orders of magnitude larger than the β-decay of tritium. Therefore, the β-decay electron could not have been in the nucleus, or else we would've observed a decay on the order of GeV

Question 5: Kinematics in 2-body particle decays

Learning objectives

In this question you will:

- Derive a relativistic expression that we will use often during the semester

For the decay $a \to 1 + 2$, show that the mass of particle $a$can be written:

$$m_a^2 = m_1^2 + m_2^2 + 2E_1E_2\left(1 - \beta_1\beta_2\cos\theta\right)$$

where β_1 and β_2 are the velocities of the particles and θ is the angle between them

Energy-momentum relation:

$$m_a^2 = E_a^2 - \overrightarrow{p_a}^2$$

Conservations of energy and momentum

$$E_a = E_1 + E_2, \overrightarrow{p_a} = \overrightarrow{p_1} + \overrightarrow{p_2}$$

plug into E-m relation:

$$m_a^2 = (E_1 + E_2)^2 - (\overrightarrow{p_1} + \overrightarrow{p_2})^2$$
$$m_a^2 = E_1^2 + E_2^2 + 2E_1E_2 - \overrightarrow{p_1}^2 - \overrightarrow{p_2}^2 - 2\overrightarrow{p_1} \cdot \overrightarrow{p_2}$$
$$m_a^2 = (E_1^2 - \overrightarrow{p_1}^2) + (E_2^2 - \overrightarrow{p_2}^2) + 2E_1E_2 - 2\overrightarrow{p_1} \cdot \overrightarrow{p_2}$$

By E-M relation again,

$$m_a^2 = m_1^2 + m_2^2 + 2E_1E_2(1 - \frac{\overrightarrow{p_1} \cdot \overrightarrow{p_2}}{2E_1E_2})$$

Because

$$E = \gamma m, \vec{p} = \gamma m \beta$$
$$\therefore m_a^2 = m_1^2 + m_2^2 + 2E_1E_2\left(1 - \beta_1\beta_2 \cos\theta\right)$$

```
In [ ]:
```

```
In [ ]:
```