Universität Zürich

Dalitz plot

Silvio Donato
(silvio.donato@cern.ch)

PHYS451 - Experimental Particle Physics
Exercise class 3
4th October 2016

http://sdonato.web.cern.ch/sdonato/UZH

Exercise 4

- What are the particle in the final state of this experiment?
- Have you found any three-body resonance?
- What is the Dalitz plot?
  - Is there any symmetry? Why?
  - Have you found any two-body resonance?
- (Bonus) Measure the two-body resonance mass.

```
from ThreeBodyDecay import exp4
(part1,part2,part3) = exp4()
print type(part1) #TLorentzVector
help(part1)
```

Exercise 5

- The $\pi^0 \rightarrow \gamma\gamma$ in a very short time. Its mass is 140 GeV.
  - Plot the previous Dalitz plot.
- (Bonus) Improve the mass resolution exploiting the π^0 mass. What is the effect on the three-body invariant mass?

```
from ThreeBodyDecay import exp5
(part1,part2,part3,part4,part5,part6) = exp5()
print type(part1) #TLorentzVector
help(part1)
```

Part II

Simulate a 3-body decay

Divide et impera

- The 3-body decay is a quite complex process.
- To solve it we:
  - split the problem in many simpler problems
- In the next slides we will simulate the Ex.1 problem:
  - $D_s^+ \rightarrow \phi(1020) + \pi^+ \rightarrow K^+ + K^- + \pi^+$

$m(D_s^+) = 1968$ MeV, $m(\pi^+) = 140$ MeV, $m(K^+) = 493$ MeV.

Divide et impera

- The 3-body decay with resonances is two 2-body decays.
- The two body decay is a two body decay at rest + boost
- The two body decay at rest of D_s^+ and ϕ is isotropic and with fixed energies.

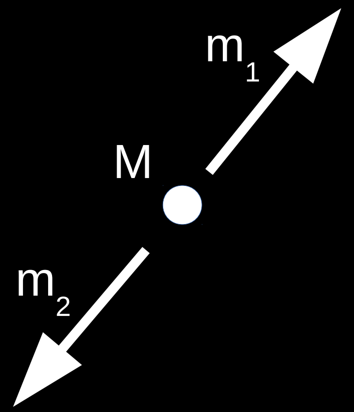
- Eventually, we add background and resolution effects.

Exercise 1

- Write a function that simulate the decay of a particle with mass M at rest into two particles with mass m_1 and m_2.
- Checks:
  - Is the invariant mass of the pair of particles in the final states equal to the original particle mass M?
  - Are the final state particles distribuited isotropically?

Exercise 2

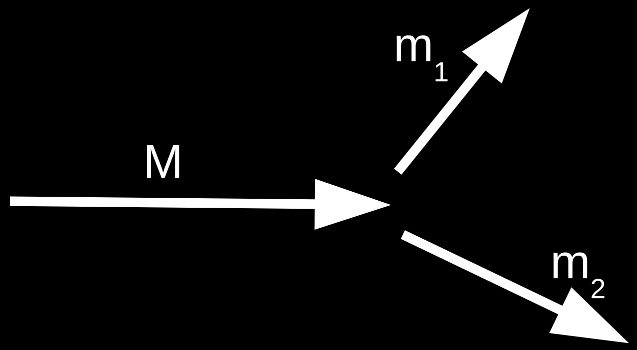
- Write a function that simulate the decay of a particle with mass M into two particles with mass m_1 and m_2.
- Checks:
  - Is the invariant mass of the pair of particles in the final states equal to the original particle mass M?
  - Are the final state particles distribuited isotropically?




```
def decayRestFrame(M,m1,m2):
    part1 = TLorentzVector()
    part2 = TLorentzVector()
    [...]
    return part1,part2
```

$$|\vec{p}_1| = |\vec{p_2}| = \frac{[(M^2-(m_1+m_2)^2)(M^2-(m_1-m_2)^2)]^{1/2}}{2M}$$

Exercise 3



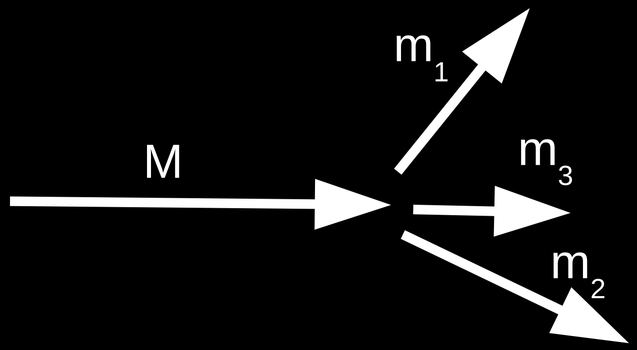
- Write a function that simulate the decay of a particle at rest with mass M into two particles with mass m_1 and m_2.
- Checks:
  - Are the energy/momentum conserved?

```
def decay(part,m1,m2):
    M = part.M()
    part1,part2=decayRestFrame(M,m1,m2)
    [...]
    return part1,part2
```

```
p = TLorentzVector(0,0,10,100)
boost = part.BoostVector()
p.Boost(-boost)
# p is now in the rest frame
```

Exercise 4

- Write a function that simulate the decay of a particle with mass M and pZ into two particles with mass m_1 and m_A, and the second one decays into m_1 and m_2.
- Checks:
  - Are the energy/momentum conserved?

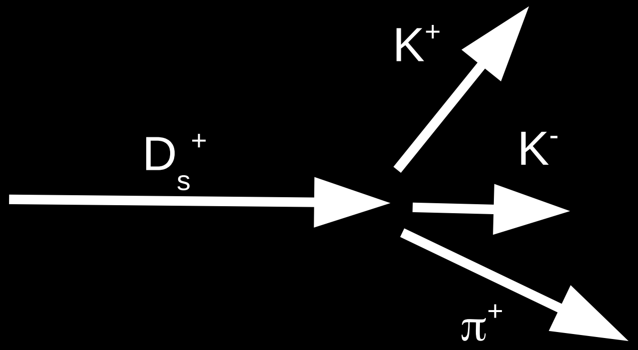



```
def generateEvent(M, m1, m2, m3, pZ=10, mA):
[…]
return part1,part2,part3
```

```
p = TLorentzVector(0,0,10,100)
boost = part.BoostVector()
p.Boost(-boost)
# p is now in the rest frame
```

Exercise 5

- Draw the Dalitz plot for
  - $D_s^+ \to \overline{K}^{*0+}(892) \quad + K^+ \to K^+ + K^- + \pi^+$ (33%)
  - $D_s^+ \to \phi(1020) \quad + \pi^+ \to K^+ + K^- + \pi^+$ (33%)
  - $D_s^+ \to K^+ + K^- + \pi^+$ [no resonance] (33%)



Exercise 6

- Smear the particle momenta taking into account an angular resolution of ~1% and an energy resolution of ~10%.
- How does the Dalitz plot change? What is the dominant error?
  - Test other values.

```
rnd = TRandom3()
for i in range(1000):
        x = rnd.Gaus()
        #x is distributed as a Gaussian distribution
        #with mean 0 and sigma 1
```

Exercise 7

- Add a non-resonant background with the three-particle invariant mass distributed as a decreasing exponential

```
rnd = TRandom3()
for i in range(1000):
	x = rnd.Exp(-2.5)
	#x is distributed as f(x) = exp(-2.5*x)
```