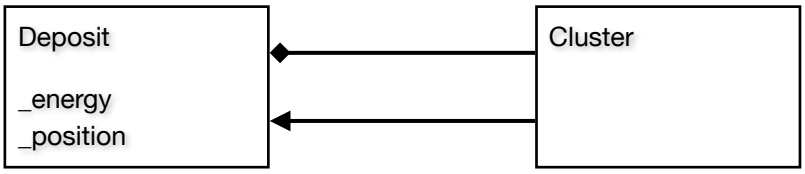
Computing Methods for Physics
23 January 2019

You must submit your exam by following the instruction at http://www.roma1.infn.it/people/rahatlou/cmp/

Composite Pattern for Calorimetric Jets (15 pt)

Calorimetric jets are used to evaluate the energy of quarks and gluons emitted in hadronic interactions. An energy calorimeter is typically divided in cells, each measuring an energy and providing a position. Energy deposits in the electromagnetic and hadron calorimeters are combined into clusters. Clusters are then combined together to form the jets.

Implement the following composite pattern



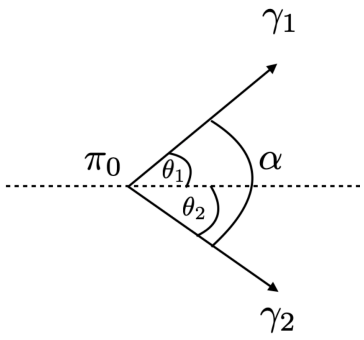
1. Class **`Deposit`** must have two data members: **`_energy`** ad **`_position`**
    a. Use **`Vector3D`** (developed during the course) for position. You can use a simple float instead (-3 penalty).
2. Overload **`+`** and **`+=`** operators for **`Deposit`**. NB: adding two **`Deposit`**s does NOT return a **`Cluster`**.
3. In class **`Cluster`** implement **`add`** and **`remove`** methods
4. Position of a cluster is computed as the weighted average position of its individual deposits. Deposits with more energy have a higher weight.
5. Overload the **`<<`** operator properly for both classes to print info about the energy and position. For a cluster, it must also print the number of children contained in the cluster
6. Test your classes with a test program **`app.cc`** .

Provide **`{Deposit,Cluster,Vector3D}.{hh,cc}`** for evaluation. Submission of **`app.cc`** is not mandatory. You will be asked to write a test application during the oral discussion to test your classes.

Evaluation will be based on: successful compilation, correct use of C++ syntax, return type and arguments of functions, data members and interface of classes, unnecessary void functions, use of unnecessary C features, correct mathematical operations, and correct physics

COMPUTING METHODS FOR PHYSICS
23 JANUARY 2019

Angular resolution with python (15 pt)



The neutral pion with mass of 135 MeV decays almost exclusively in 2 photons. The photons are emitted with angles θ_1 and θ_2 with respect to the direction of the pion, ad the opening angle is given by $\alpha = \theta_1 + \theta_2$ and $0 \leq \alpha \leq \pi$.

The decay probability as function of the opening angle is

$$f(\alpha) = \frac{m_0^2}{4E_0^2} \frac{cos(\alpha/2)}{sin^2(\alpha/2)\sqrt{sin^2(\alpha/2) - m_0^2/E_0^2}}$$

where E_0 is the energy of the decaying π_0. Provide a plot of this probability as a function of α for E_0 = 0.5, 1, 5, 10, 50, 100 GeV. The plot must show the curves for different energies with different colours and legends.

The detector can distinguish the two photons when $\alpha > \alpha_0$=0.2 rad.

Simulate 10000 π_0 decays for each value of E_0 = 0.5, 1, 5, 10, 50, 100 GeV.

Implement a function for integration with your favorite method (e.g. midpoint or MC). Compute the fraction of decays with resolved (separated) photons

$$p(E_0) = \frac{\int_{\alpha_0}^{\pi} f(\alpha) d\alpha}{\int_0^{\pi} f(\alpha) d\alpha}$$

for each energy and plot it as function of E_0. This curve is valid only for an ideal detector with perfect angular resolution. In order to account for the detector resolution apply a Gaussian smearing.

1. For each decay generate randomly the true values of θ_1 and θ_2 making sure that $0 \leq \alpha \leq \pi$
2. Apply a Gaussian smearing to θ_1 and θ_2 with mean of μ=0 and width σ.
3. Recompute the fraction $p(E_0)$ for different values of the Gaussian width σ = 1%, 5% and 10%.
4. Make a new plot of $p(E_0)$ versus E_0, and overlay the 4 scenarios: no smearing (ideal), 1%, 5% and 10% smearing.
5. Compute the fraction of decays that are lost (photons not resolved) when we go from the ideal detector to 10% angular resolution, and print it on the screen.

Evaluation will be based on use of python features and data structures, comprehensions (instead of C-style for loops), NumPy objects, labels, units, clarity ad correctness of plots.