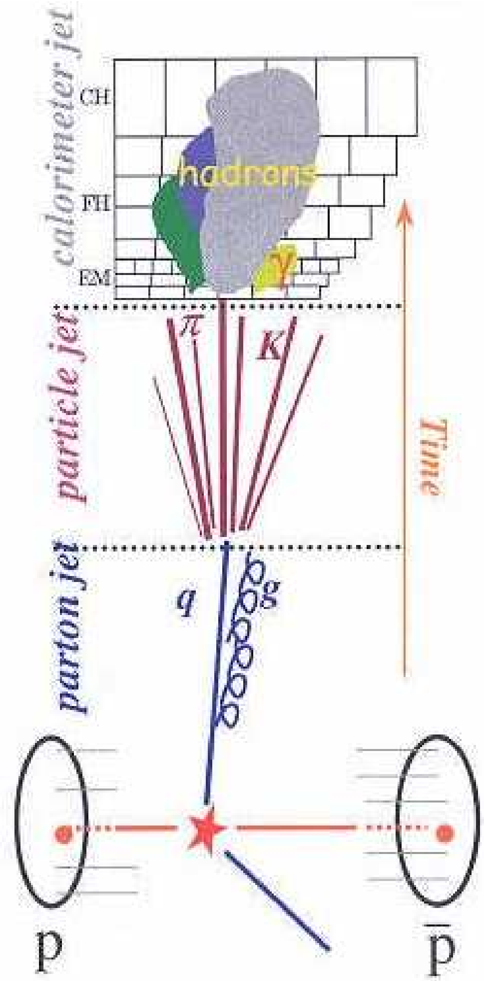
Jet Algorithmen in CDF

Yves Kemp

Universität Karlsruhe
Institut für Experimentelle Kernphysik

2. April 2001

Woraus besteht ein Jet?

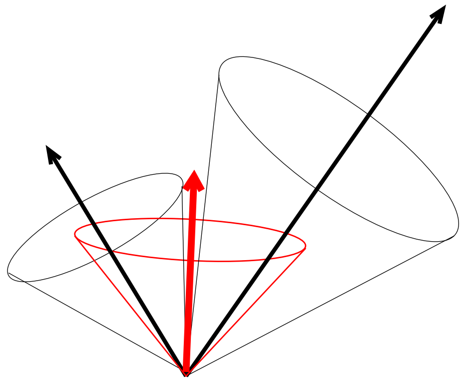


1. Partonen Jet: nur Quarks und Gluonen
2. Teilchen Jet: Teilchen im Endzustand (nach Hadronisation)
3. Kalorimeter Jet: Im Detektor gemessene Objekte

Mit der Annahme $m \approx 0$ braucht man für jedes Teilchen (Kalorimeterobjekt) $\vec{P}$ oder η, φ und $E_T (= P_T)$

Anforderungen an einen Jet Algorithmus

1. Infrarot-sicher:



2. Kollinear-sicher:

   - Resultate unabhängig der Seed-Ordnung
   - Kein Einfluß von Aufteilung eines Impulses auf mehrere Detektorzellen

3. Boost-Invariant
4. Unabhängig vom Detektor
5. Unabhängig von den betrachteten Objekten (Partonen, Hadronen im Endzustand oder Detektor-Objekten)

Punkt 1, 2 und 3 sollten von jedem Algorithmus erfüllt werden
Punkt 4 und 5 sollten näherungsweise erfüllt sein.

Cone Algorithmus

(Weiterentwicklung des Snowmass-Algorithmus)

Erste Variante: seed P_T order

1. Ordne alle Teilchen mit $P_T > 0.5$ GeV und $|\eta| < 1$ nach P_T
2. Teile alle Teilchen innerhalb $\sqrt{(\Delta\eta)^2 + (\Delta\varphi)^2} = R = 0.7$ innerhalb des ersten Eintrages einem Jet zu
3. Lösche alle Teilchen aus diesem Jet
4. Wiederhole bis alle Teilen einem Jet zugeteilt sind

Die kinematischen Größen des Jets sind dann wie folgt definiert (E-Scheme):

$$\begin{aligned} \widetilde{P}_T^{\text{jet}} &= \left(E^{\text{jet}}, \vec{P}^{\text{jet}}\right) = \sum_{\Delta R_i < 0.7} \left(E^i, P_x^i, P_y^i, P_z^i\right) \\ &\rightarrow \quad P_T^{\text{jet}}, y^{\text{jet}}, \varphi^{\text{jet}} \end{aligned}$$

Cone Algorithmus, Varianten

Zweite Variante: seed $\sum P_T$ order

Anstatt die "seeds" nach P_T zu ordnen, ordnet man sie nach der skalaren Summe von P_T aller in $R < 0.7$ enthaltenen Teilchen.

Dritte Variante: Ungewichteter midpoint Algorithmus

Einfügen von "pseudo-seeds" im Mittelpunkt von jeweils zwei seeds.
Ansonsten wie vorherige Variante.

Vierte Variante: Gewichteter midpoint Algorithmus

Wenn $0.7 < \Delta R < 1.4$ zwischen zwei seeds:
Einfügen von pseudo-seed im P_T-gewichteten Mittelpunkt
Ansonsten wie Variante 2.

Fünfte Variante: Seedless Cone Algorithmus

k_T Algorithmus: Preclustering

k_T Algorithmus: $\mathcal{O}(n^3)$
$\rightarrow$ Zusammenfassen von mehreren Teilchen zu einem Prejet da sonst die Rechenzeit zu groß ist.

1. Für jeden Kalorimeter-tower: P_T aus $P_{T,EM}$ und $P_{T,HAD}$
2. Ordne die Kalorimeter-tower mit $P_T \gtrsim 100$ MeV nach η
   Erster Eintrag in der Liste ist Precluster
3. Ordne alle Einträge in $R < 0.2$ diesem Precluster zu, lösche die Einträge und kombiniere zu neuem Precluster
4. Wiederhole bis alle Tower Precluster sind

k_T Algorithmus

(Catani, Dokshitzer, Seymour, Webber, Ellis, Soper)

1. Ordne jedem Precluster d_i und jedem Precluster-Paar d_{ij} wie folgt zu:

$$\begin{aligned} d_i &= P_{T,i}^2 \\ d_{i,j} &= \min\left(P_{T,i}^2, P_{T,j}^2\right) \frac{(\eta_i - \eta_j)^2 + (\varphi_i - \varphi_j)^2}{D^2} \end{aligned}$$

   wobei $D = 0.7$ vergleichbar mit R aus dem Cone-Algorithmus ist.
2. Suche Minimum der d_i und $d_{ij} \rightarrow d_{\text{min}}$
3. Falls d_{min} ein d_{ij} ist, fasse Precluster i und j zusammen
4. Falls d_{min} ein d_i ist, ist Precluster ein Jet.
5. Wiederhole bis alle Precluster in einem Jet sind oder bis $d_{\text{min}} < d_{\text{min}}^{\text{cut}}$

Mit dem Parameter $d_{\text{min}}^{\text{cut}}$ kann man also Jet-Multiplizität steuern

Zusammenfassung und Ausblick

Kritische Punkte der Cone Algorithmen:

- Merging
- Infrarot-Sicherheit bei der ersten Variante
- Kollinear-Sicherheit bei der ersten Variante
- Probleme bei nicht-kreisförmigen Jets
- Reihenfolge der seeds

Kritische Punkte des k_T-Algorithmus:

- Kalibrieren
- Preclustering

Empfehlungen der Run II jet physics group:

- CDF und DØ gleiche Version von sowohl einem Cone als auch einem k_T-Algorithmus während Run II
- Bevorzugter Bebrauch von Vierer-Vektoren gegenüber abgeleiteten Variablen (η, φ, E_T)