

Camille Bélanger-Champagne



UPPSALA
UNIVERSITET

Jet Algorithms at Hadron Colliders

QCD@Colliders - Presentation
January 15th, 2007



OUTLINE

- General requirements for jet algorithms
- Jet kinematic variables
- Cone algorithms
- Cluster algorithms
- Jet energy profile
- Summary



General Requirements I

- **Efficient at finding all physically relevant jets as described by perturbative QCD.**
- **Well-behavedness:**
 - Collinear safe: finds the same jets given one particle or two collinear particles with same total energy.
 - Infrared safe: finds the same jets even in the presence of soft radiation in the event.
- **Full specification: includes jet definition, clustering prescription, splitting and merging, ...**



General Requirements II

- Does not depend on details of the detector geometry.
- Behaves reasonably with increased luminosity (multiple scatterings, high detector occupancy): jet sizes stay under control.
- Invariant under boosts (more important for hadron collisions than e^+e^-).
- Order independant: behaves equally at parton level, particle level and detector level.



General Requirements III

- Technical requirements:
 - Easy to calibrate.
 - Easy to use in various experimental setups.
 - CPU efficient.
 - Does not increase detector energy smearing or angular biases.

Most experiments at hadron colliders (e.g. Tevatron) use mainly cone algorithms for jet reconstruction.



Jet Kinematic Variables I

- Jets can be made up of partons, particles or, in the case of real experimental data, calorimeter cells and towers.
- The values of the kinematic variables associated with the jets are a combination built from the ones associated with the individual components (partons, particles, calorimeter towers or cells).
- Multiple combination schemes exist.

Jet Kinematic Variables II

- Original Snowmass scheme (where i is a parton, particle, calorimeter object):

$$E_T^{Jet} = \sum_{i \in Jet} E_T^i$$

$$\eta^{Jet} = \frac{1}{E_T^{Jet}} \sum_{i \in Jet} E_T^i \eta^i$$

$$\phi^{Jet} = \frac{1}{E_T^{Jet}} \sum_{i \in Jet} E_T^i \phi^i$$

- Advantages: simple definition, simple transformations under boosts, accurate approximation for small jet masses
- Disadvantages: E_T not a true energy creating difficulties with kinematic boundaries and resummations

Jet Kinematic variables III

- At Tevatron Run II, uniform recommendation:

$$p^{Jet} = (E^{Jet}, \mathbf{p}^{Jet}) = \sum_{i \in Jet} (E^i, p_x^i, p_y^i, p_z^i)$$

$$p_T^{Jet} = \sqrt{(p_x^{Jet})^2 + (p_y^{Jet})^2}$$

$$y^{Jet} = \frac{1}{2} \ln \frac{E^{Jet} + p_z^{Jet}}{E^{Jet} - p_z^{Jet}}$$

$$\phi^{Jet} = \tan^{-1} \frac{p_y^{Jet}}{p_x^{Jet}}$$

- However, for simplicity, during jet reconstruction, Snowmass scheme still used. At analysis level, this definition can be used.



Cone Algorithms I

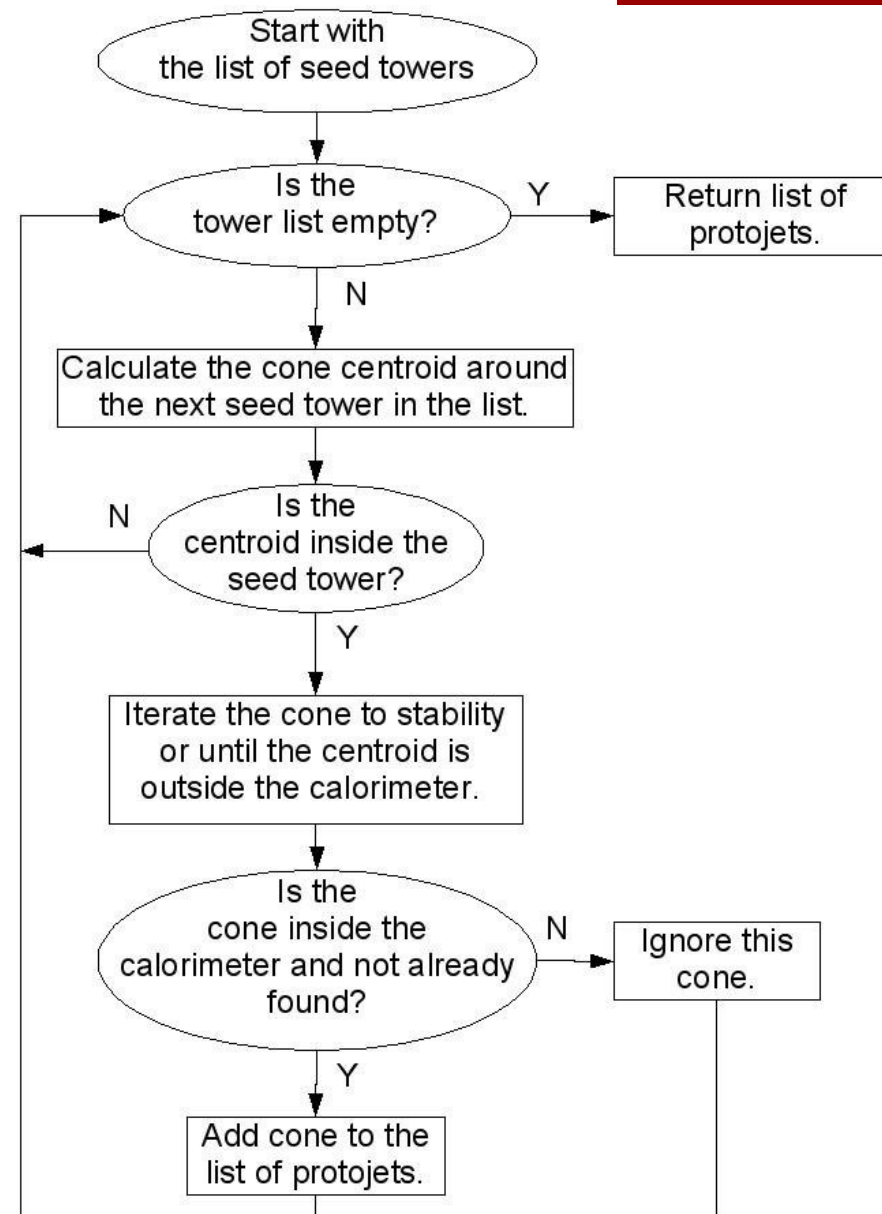
- Jets are defined as cones of fixed radius R in $\eta \times \phi$ space, where

$$R = \sqrt{\eta^2 + \phi^2}$$

- Cone formation can proceed through seeded or unseeded algorithms.
- As cones formed can overlap, a splitting and merging of overlapping cones must be included in the algorithm.

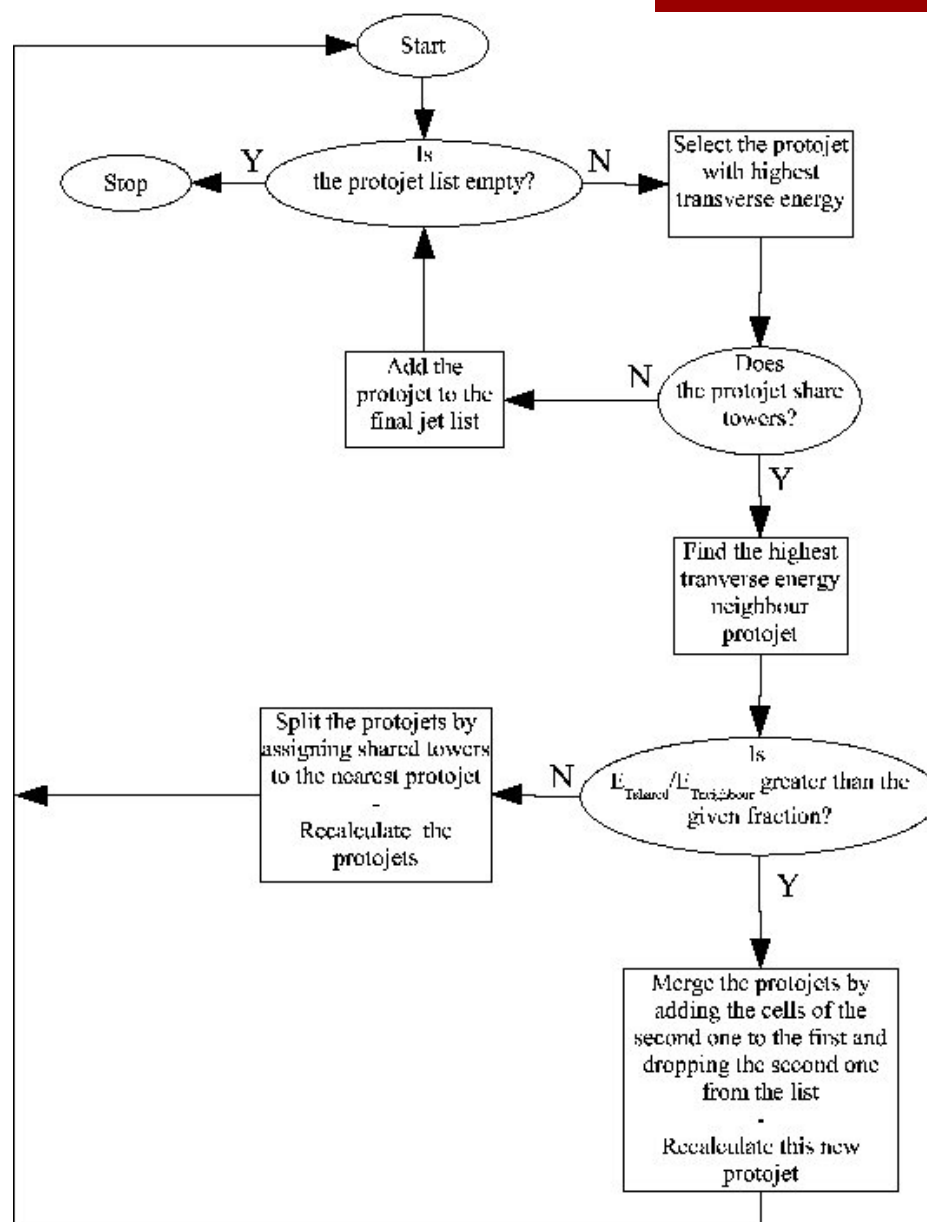
Cone Algorithms II

- Seeds are calorimeter towers above a given energy threshold.
- Seeded cone algorithms only use those seed towers as starting points for jet reconstruction.
- Advantage: speeds up computation.
- Disadvantage: not collinear or IR safe.



Cone Algorithms III

- Protojets obtained in cone formation step must be checked for overlap.
- In the presence of an overlap, cones are merged or split.
- Note that this means the final jets may not all be of radius R anymore.





Cone Algorithms IV

- Improved Legacy Cone Algorithm:
 - DØ developed for Run II.
 - A midground between the flaws of the seeded cone algorithm and the CPU consumption of the unseeded cone algorithm.
 - Use towers above a given threshold as seeds. Add to list of seeds the midpoint between a seed i and all other seeds j, k, l, \dots if the distance ΔR between i and j satisfies $\Delta R < 2.0 * R_{cone}$.
 - Approaches the performance of the seedless algorithm.

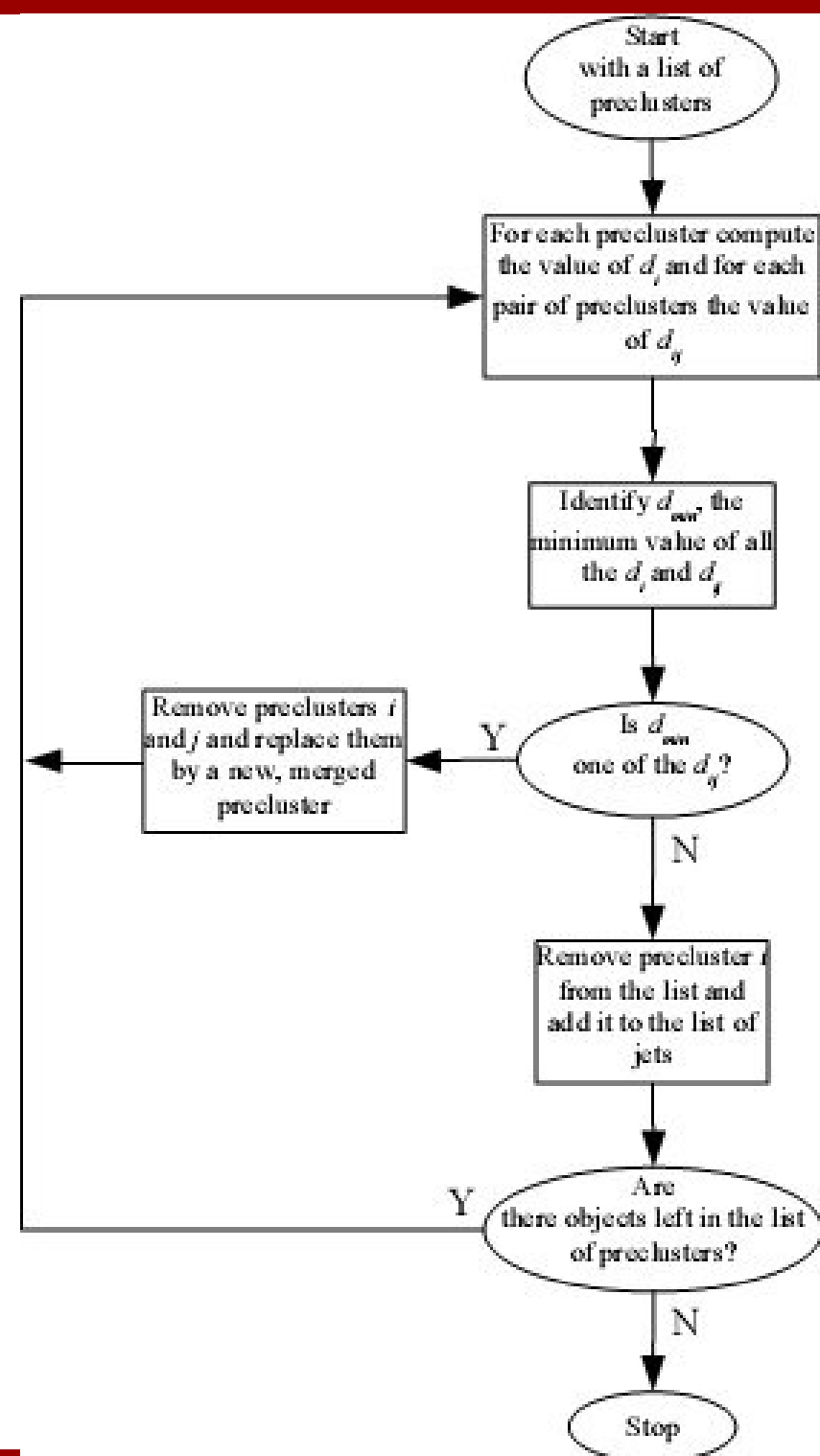
k_T Algorithm I

- Clustering algorithm most commonly used at hadron colliders.
- For all energy clusters define:

$$d_i = p_{T,i}^2$$
$$d_{ij} = \min(p_{T,i}^2, p_{T,j}^2) \frac{(y_i - y_j)^2 + (\phi_i - \phi_j)^2}{D^2}$$

where $D \sim 1$ is a parameter of the algorithm.

- d_{ij} is the minimal relative transverse momentum between i and j .



k_T Algorithm II

- Two distinct approaches to stopping the clustering in the algorithm.
- *Catani et al.*, introduce parameter d_{cut} to define the scale of the hard process a priori or on an event-by-event basis. When $d_{min} > d_{cut}$ clustering stops. Jets with $p_T^2 < d_{cut}$ are “beam jets”.
- *Ellis & Sopper* keep merging clusters until all remaining are separated by:

$$(y_i - y_j)^2 + (\varphi_i - \varphi_j)^2 > D^2$$

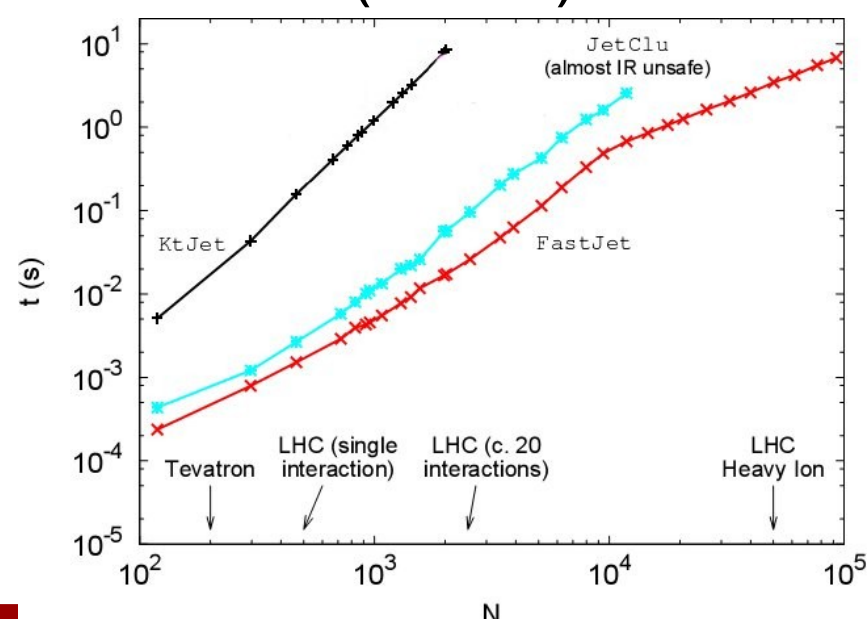
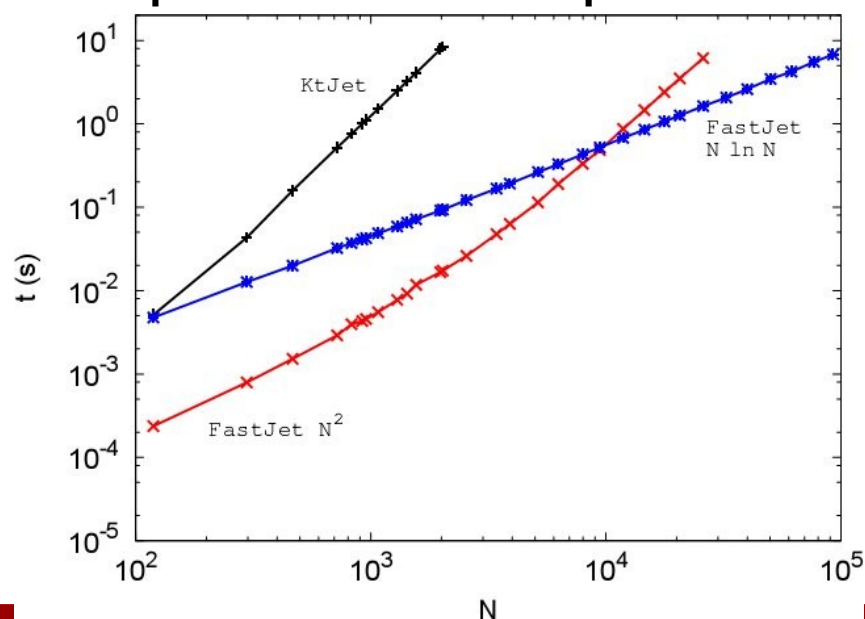


k_T Algorithm III

- Computing requirements traditionally $O(n^3)$ for this algorithm.
- At Tevatron, used a preclustering step to limit the number of input preclusters to the algorithm.
- At CDF, take only calorimeter towers above threshold $\sim 100\text{MeV}$.
- At DØ, must take into account the frequent presence of cells of negative energy. From an η -ordered list of towers, group towers spatially close in clusters of 0.2 in $\eta \times \phi$ space. Redistribute towers from preclusters with negative energy or energy below 200MeV to neighbouring clusters.

k_T Algorithm IV

- New work from *Cacciari and Salam* indicates a k_T algorithm could be implemented to be $O(N \ln N)$, which would allow it to be used extensively at LHC (and at Tevatron).
- Does so using known properties of nearest neighbours to avoid having to build a full table of all possible d_{ij} (then $O(N^2)$), which in turn makes the problem into known and highly optimized computer science problems which have $O(N \ln N)$ solutions.





Jet Energy Profile I

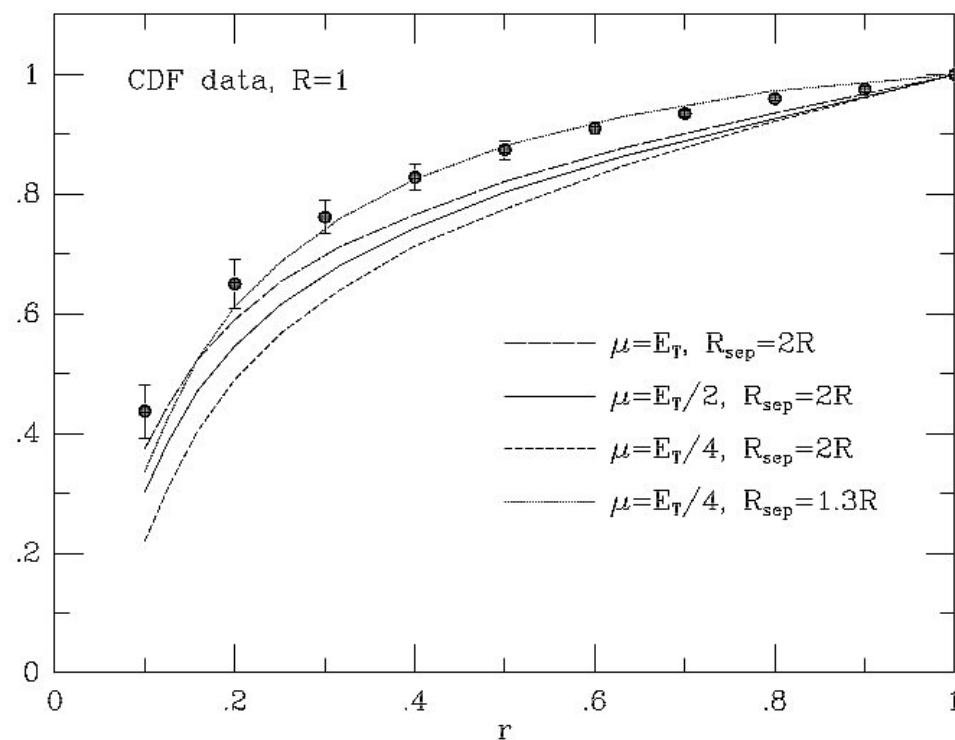
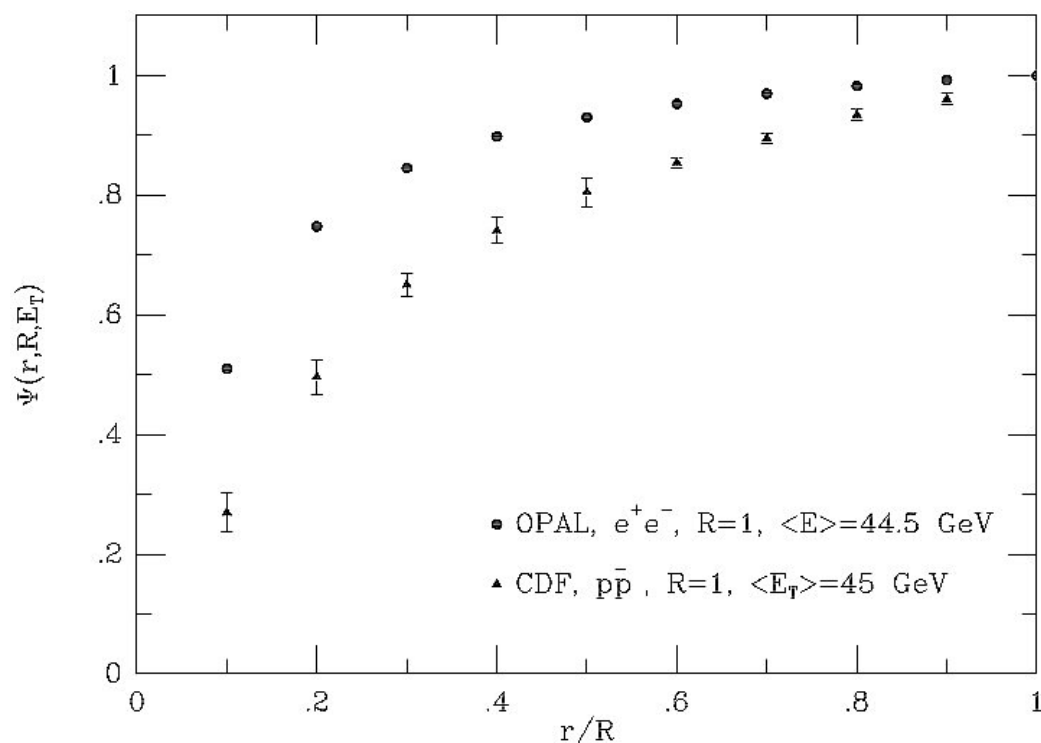
- A description of the transverse energy flow within the jet.
- At QCD NLO, the jet energy profile is modeled by a pair of partons. NLO is the first order where the profile depends on cone size: it constitutes an LO energy profile prediction.
- Consequences: strong renormalization scale dependence, dependence of jet profile on jet algorithm details at $O(\alpha_s^3)$, no existing complete quantitative energy profile description.

Jet Energy Profile II

- The fractionnal E_T profile is given by:

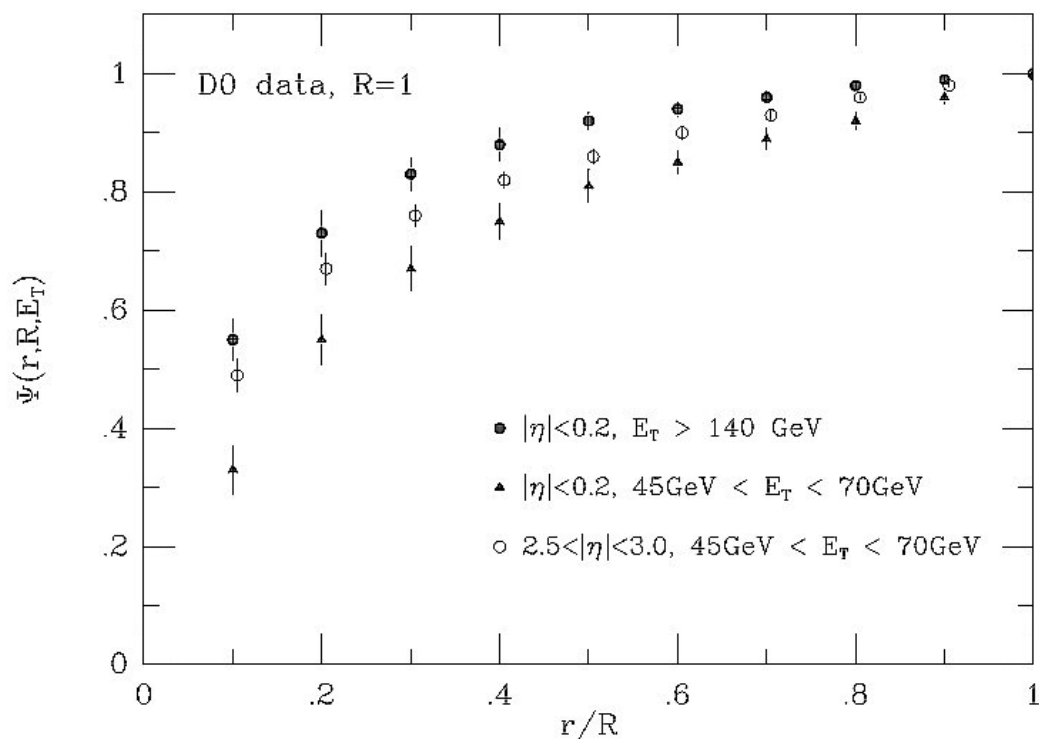
$$\Psi(r, R, E_T) = \frac{(\text{Transverse energy within cone of radius } r)}{(\text{Transverse energy within cone of radius } R)}$$

- where $r < R$ and the two cones are collinear.



Jet Energy Profile III

- For a given value of r , the energy profile function Ψ increases with E_T , indicating that the jets are narrower as more of their energy is located near the center of the cone.

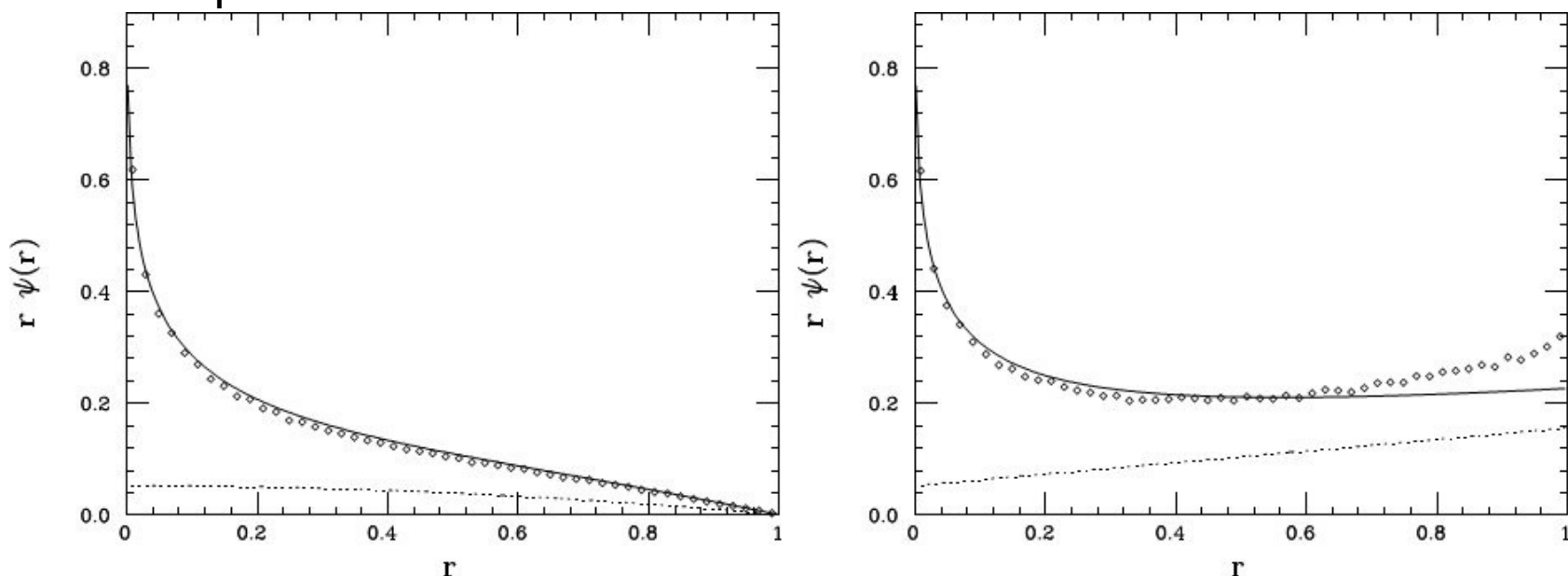


Jet Energy Profile IV

- Can also define a differential energy profile:

$$\psi(r, R, E_T) = \frac{d\Psi(r, R, E_T)}{dr}$$

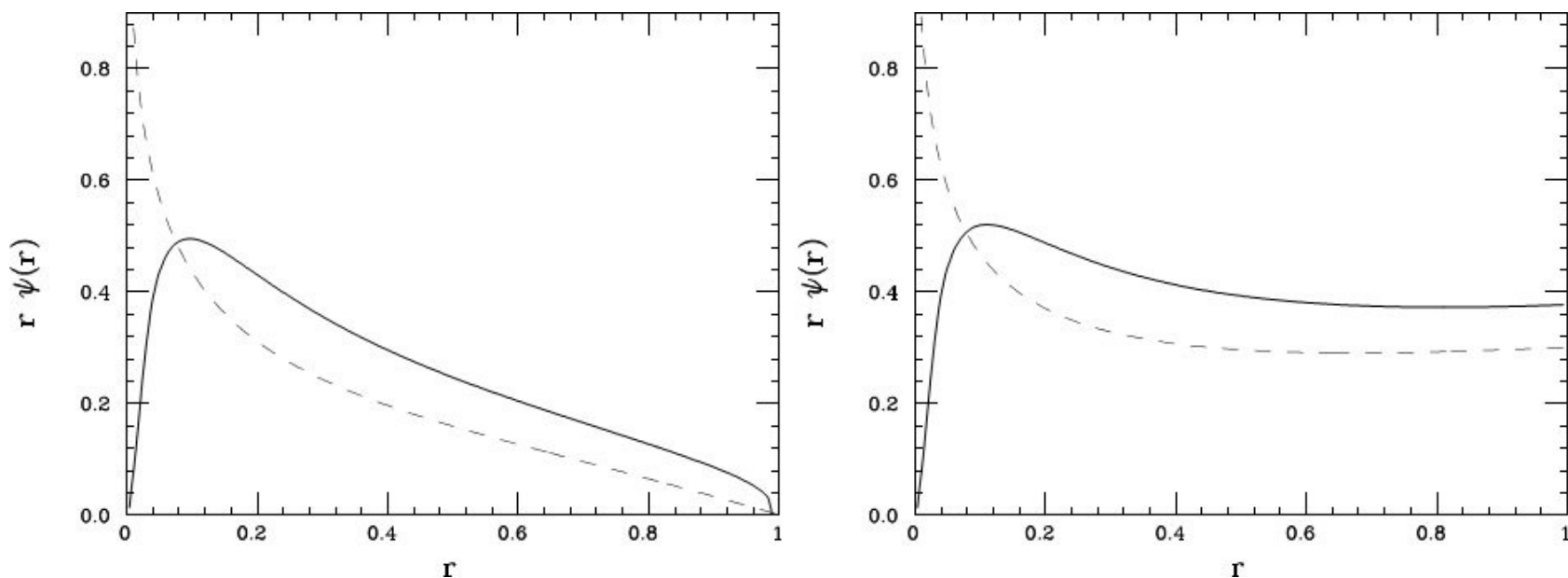
- The energy profile depends on the algorithm used (k_T left, midpoint cone right):



Points are the exact tree-level ME calculation, curve is analytical formula, 50GeV jets.

Jet Energy Profile V

- Some work done by Seymour on NLO corrections to QCD jet profile prediction (running coupling, resummation ($\sim \alpha_s^n \ln^{2n-1} r$), power corrections):



k_T (left) and midpoint cone (right). LO (dashed) and LO+all NLO corrections (solid) for 50 GeV jets.



Summary

- Jet algorithm design is particularly challenging at hadron and heavy ion colliders because of the high occupancy of the detectors.
- For historical reasons, and because of their simplicity and speed, cone algorithms have mostly been used.
- Cluster algorithms fulfill, in principle, more of the requirements of the ideal jet reconstruction algorithm.
- New developments in cluster algorithm design lead to think that they could start being used at all levels in LHC experiments.
- The energy profile is an interesting quantity to describe jets. As it currently stands, there is no satisfying NLO QCD description of the energy profile function.



References

- Course Textbook.
- *Blazey et al.*, hep-ex 0005012.
- *Busato and Andrieu*, DØNote 4457.
- *Cacciari and Salam*, hep-ex 0512210.
- *Ellis, Huston and Tönnesmann*, hep-ph 0111434.
- *Seymour*, hep-ph 9707338.
- *Seymour*, LU TP 93-8, Z.Phys.C62:127-138, 1994.