Jet finding Algorithms at Tevatron

B.Andrieu (LPNHE, Paris)

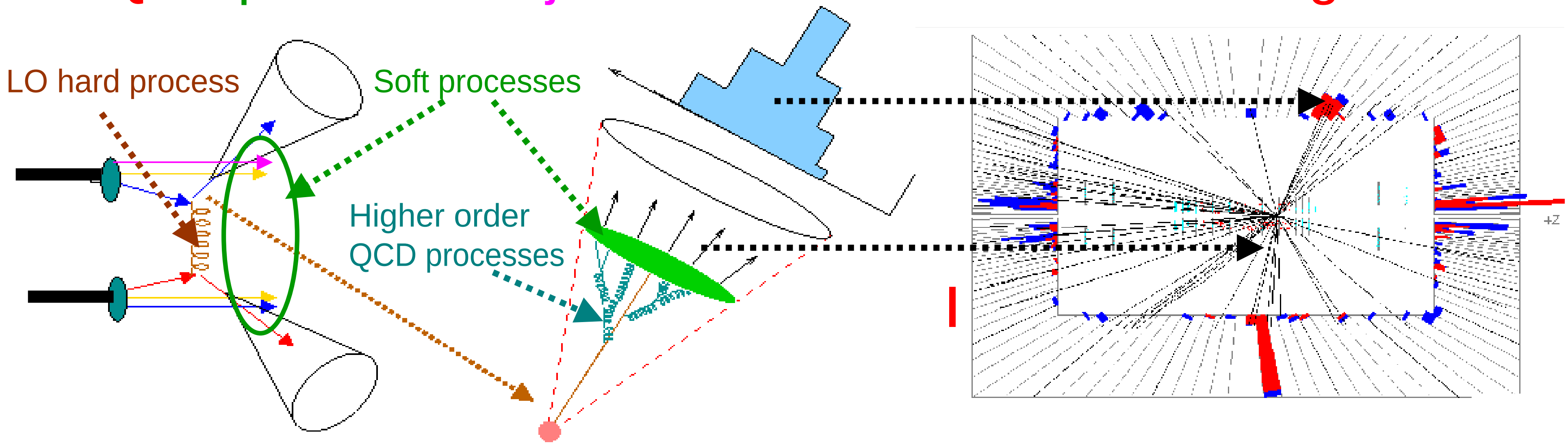
On behalf of the DØ collaboration

Outline:

- Introduction
- The Ideal Jet Algorithm
- Cone Jet Algorithms: RunII/RunI, DØ/CDF
- $k_\perp$ Jet Algorithm
- Summary

Jets: from parton to detector level

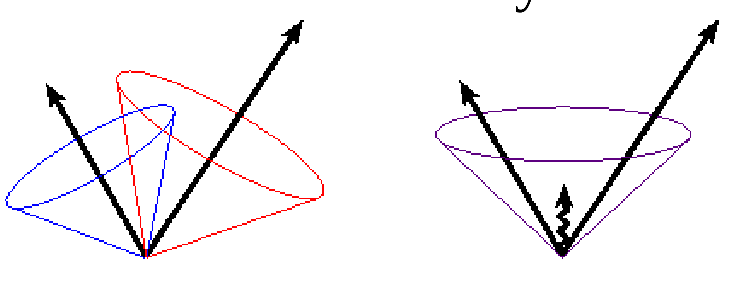
QCD partons → jets of hadrons → detector signals



QCD ⇒ { Quark and gluon jets (identified to partons) can be compared to detector jets, if jet algorithms respect collinear and infrared safety (Sterman&Weinberg, 1977)

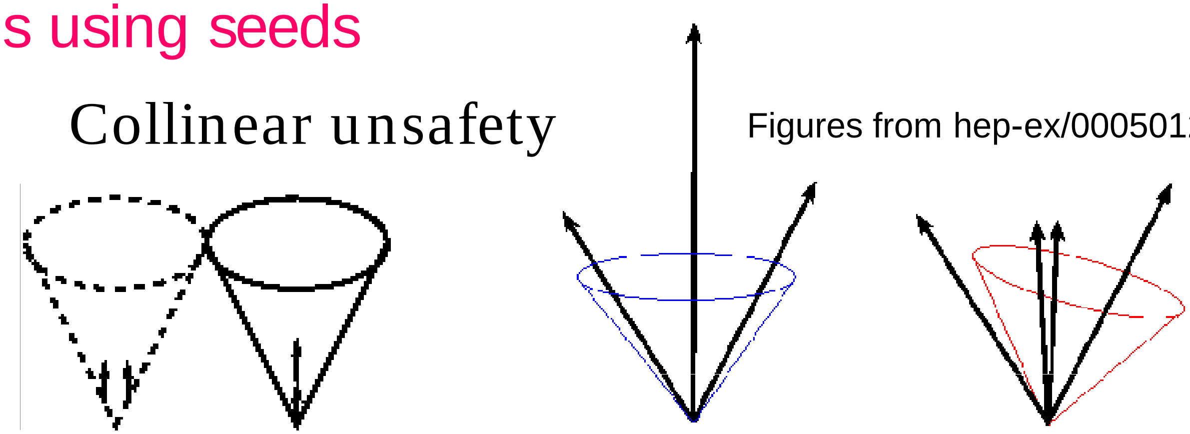
Problems of Cone Jet Algorithms using seeds

Infrared unsafety



Collinear unsafety

Figures from hep-ex/0005012



Jet definition

- Associate "close" to each other "particles"
  → **Clustering (Jet Algorithm)**
  - "particles"
    - partons (analytical calculations or parton showers MC)
    - "hadrons" = final state particles (MC particles or charged particles in trackers)
    - towers (or cells or preclusters or any localized energy deposit)
  - "close" ? → Distance
    → $\Delta R = \sqrt{\Delta\eta^2+\Delta\phi^2}$ or $\sqrt{\Delta Y^2+\Delta\phi^2}$ (preferred in RunII) for Cone Algorithm
    → relative p_T for $k_\perp$ algorithm
- Calculate jet 4-momentum from "particles" 4-momenta
  → **Recombination scheme**
  - invariant under longitudinal boosts
    → Snowmass scheme (RunI): E_T-weighted recombination scheme in (η,ϕ)
    → covariant or E-scheme (preferred for RunII): 4-momenta addition
  - used at the end of clustering but also during clustering process
    (not necessarily the same, still preferable)

The Ideal Jet Algorithm for $p\bar{p}$

Compare jets at the parton, hadron and detector level

⇨ **Jet algorithms should ensure**

General

- infrared and collinear safety
- invariance under longitudinal boosts
- fully specified and straightforward to implement
- same algorithm at the parton, hadron and detector level

Theory

- boundary stability (kinematic limit of inclusive jet cross section at $E_T = \sqrt{s}/2$)
- factorisation (universal parton densities)

Experiment

- independence of detector detailed geometry and granularity
- minimal sensitivity to non-perturbative processes and pile-up events at high luminosity
- minimization of resolution smearing/angle bias
- reliable calibration
- maximal reconstruction efficiency (find all jets) vs minimal CPU time
- replicate RunI cross sections while avoiding theoretical problems

Run I Cone Algorithm

- Based on Snowmass Algorithm: E_T -weighted recombination scheme in (η,ϕ)
- **Preclustering** (D∅, similar algorithm for CDF)

  Note: Tower segmentation in (η,ϕ) space: **D∅ → 0.1 X 0.1**, **CDF → 0.11 X 0.26**
  - start from seeds (= towers with p_T >1 GeV ordered in decreasing p_T)
  - cluster (and remove) all contiguous calorimeter towers around seed in a R= 0.3 cone
- **Clustering**
  - start from preclusters (ordered in decreasing E_T)
  - proto-jet candidate = all particles within R_{cone} of the precluster axis in (η,ϕ) space
    CDF: keep towers of the original precluster through all iterations (ratcheting)
  - proto-jet direction compared before/after recombination → iterate until it is stable
- **Merging/Splitting** (treat overlapping proto-jets)
  - $E_{T,1\cap 2} > f \,.\, \mathrm{Min}(E_{T,1}, E_{T,2})$ → Merge jets
  - $E_{T,1\cap 2} < f \,.\, \mathrm{Min}(E_{T,1}, E_{T,2})$ → Split jets = assign each particle to its closest jet
  - **D∅: f = 50 %, use only clusters with E_T > 8 GeV - CDF: f = 75 %**
- **Final calculation of jet variables** (modified Snowmass scheme)
  - scalar addition of **E_T (D∅)** or **E (CDF)** of particles to determine jet **E_T** or **E**
  - addition of 3-momenta of particles to determine jet direction, then (η,ϕ)
    Note: this procedure is not Lorentz invariant for boosts along beam axis
    CDF: $E_T = E \sin(\theta)$

Why new algorithms for Run II?

Run I Cone algorithms have many drawbacks

- different in DØ and CDF
- not infrared and collinear safe due to the use of seeds
  (collinear safety ensured at sufficiently large E_T : **E_T >20 GeV with p_T^{min} (seed) = 1 GeV in DØ**)
- preclustering difficult to match at parton or hadron level
- CDF ratcheting not modeled in theory
- ad-hoc parameter (R_{sep}) in jet algorithm at parton level
  (S.D. Ellis et al., PRL69, 3615 (1992))
- not invariant under boosts along beam axis

⇒ 2 new Cone Algorithms proposed for RunII

(G.C. Blazey et al., "RunII Jet Physics", hep-ex/0005012)

- Seedless Cone Algorithm
- RunII (= Improved Legacy or Midpoint) Cone Algorithm

⇒ use $k_\perp$ Algorithm (already used in RunI)

Seedless Cone Algorithm

Not really "seedless"

→ use enough seeds (all towers) to find all stable cones

Streamlined (faster) option

- form cone around seed, recalculate cone direction (Snowmass or E - scheme)
- stop processing seed if the cone centroid is outside of the seed tower
  CDF: use tower size X 1.1 in 1st step to avoid boundary problems
- iterate until cone direction after/before recombination is stable
- only miss low E_T proto-jets or stable directions within the same tower compared to normal version

→ Infrared and collinear safe

→ Probably close to Ideal for a Cone algorithm

→ Very computationally intensive

⇒ Use an approximation of Seedless Algorithm → RunII Cone

RunII Cone Algorithm (hep-ex/0005012)

How to build a valid approximation of the seedless algorithm?

- QCD calculation at fixed order N
  → only $2^N - 1$ possible positions for stable cones (p_i , $p_i + p_j$, $p_i + p_j + p_k$,...)
- Data: consider seeds used in RunI Cone algorithms as partons
  → in addition to seeds, use 'midpoints' i.e. $p_i + p_j$, $p_i + p_j + p_k$,...
- only need to consider seeds all within a distance $\Delta R < 2R_{cone}$
- only use midpoints between proto-jets (reduce computing time)
- otherwise algorithm similar to RunI

Other specifications of the suggested RunII cone Algorithm

- E-scheme recombination = 4-momenta addition
- use true rapidity Y instead of pseudo-rapidity η in ΔR
- use all towers as seeds ($p_T > 1$ GeV)
- splitting/merging: p_T ordered, f = 50 %

DØ Run II Cone Algorithm

Preclustering similar to RunI except:

- seeds = p_T ordered list of particles with p_T >500 MeV
- precluster = all particles in a cone of r = 0.3 around seed for Cone Jets with R $\geq$ 0.5
- precluster 4-momentum calculated using the E- scheme

Clustering

- seeds = p_T ordered list of preclusters
  except those close to already found proto-jets: ΔR (precluster,proto-jet)< 0.5 R_{cone}
- cone drifting until
  - cone axis coincides with jet direction
  - p_T < 0.5 Jet p_T^{min}
  - # iterations = 50 (to avoid ∞ cycles)
- remove duplicates
- repeat same clustering for midpoints* except
  - no condition on close proto-jet
  - no removal of duplicates

* for pairs only, calculated using p_T –weighted mean

Merging/splitting similar to RunI except:

- use p_T ordered list of proto-jets (from seeds and midpoints)
- at each merging/splitting
  - recalculate 4-momenta of merged/splitted jets
  - re-order list of merged/splitted jets

The Smaller Search Cone Algorithm

- Jets might be missed by RunII Cone Algorithm (S.D. Ellis et al., hep-ph/0111434)
  → low p_T jets
  - too close to high p_T jet to form a stable cone (cone will drift towards high p_T jet)
  - too far away from high p_T jet to be part of the high p_T jet stable cone
- proposed solution
  - remove stability requirement of cone
  - run cone algorithm with smaller cone radius to limit cone drifting
    ($R_{search} = R_{cone} / \sqrt{2}$)
  - form cone jets of radius R_{cone} around proto-jets found with radius R_{search}

Remarks

- Problem of lost jets seen by CDF, not seen by DØ
  → A physics or an experimental problem?
- Proposed solution unsatisfactory w.r.t. cone jet definition

⇒ DØ prefers using RunII Cone without Smaller Search Cone

$k_\perp$ Algorithm

Description of inclusive $k_\perp$ algorithm (Ellis&Soper, PRD48, 3160, (1993))

- DØ: geometrical 2x2 preclustering, remove preclusters with E < 0
- p_T ordered list of particles → form the list of $d_i = (p_T^i)^2$
- calculate for all pairs of particles, $d_{ij} = \mathrm{Min}((p_T^i)^2, (p_T^j)^2)\ \Delta R/D$
- find the minimum of all d_i and d_{ij}
  - if it is a d_i, form a jet candidate with particle i and remove i from the list
  - if not, combine i and j according to the E-scheme
  - use combined particle i + j as a new particle in next iteration
  - need to reorder list at each iteration → computing time $\propto O(N^3)$ (N particles)
- proceed until the list of preclusters is exhausted

Remarks

- originally proposed for e^+e^- colliders, then adapted to hadron colliders (S. Catani et al., NPB406,187 (1993))
- universal factorisation of initial-state collinear singularities
- infrared safe: soft partons are combined first with harder partons
- collinear safe: two collinear partons are combined first in the original parton
- no issue with merging/splitting

Summary

- RunII (Midpoint) Cone Algorithm clear improvement over RunI Algorithm
- problems or questions still open (not exhaustive list):
  - DØ uses RunII Cone (Midpoint) Algorithm (no smaller search cone)
  - CDF uses JetClu (RunI) Cone Algorithm + Smaller Search Cone Algorithm
  - differences of DØ implementation w.r.t. RunII Cone recommendations
  - usefulness of a p_T cut on proto-jets before merging/splitting at high luminosity?
  - procedure chosen for merging/splitting optimal?
  - origin of the difference DØ vs CDF for lost jets problem?
- $k_\perp$ algorithm less intuitive, but conceptually simpler and theoretically well-behaved.
  - studies needed, which should be done also for the RunII Cone Algorithm (jet masses, sensitivity to experimental effects, ...).

$\Rightarrow$ shouldn't we put more effort on using $k_\perp$ algorithm? (personal statement)