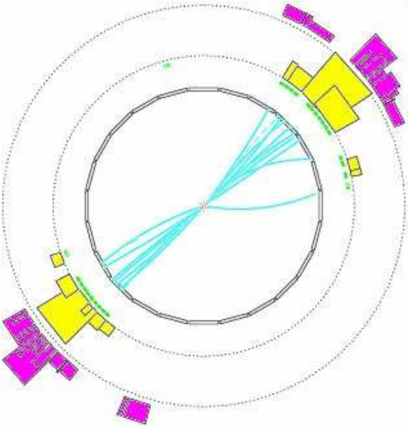
Jets at Hadron Colliders (1)

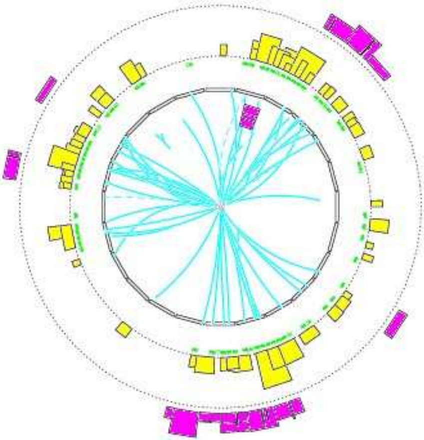
Gavin Salam

CERN, Princeton and LPTHE/Paris (CNRS)

CERN Academic Training Lectures
30 March - 1 April 2011

Jets

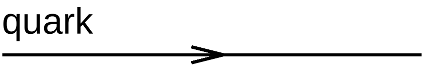




Jets are everywhere in QCD
Our *window on partons*

But *not* the same as partons:
Partons ill-defined; jets *well-definable*

Why do we see jets? Parton fragmentation



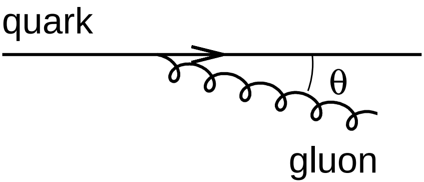
Gluon emission:

$$\int \alpha_s \frac{dE}{E}\frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



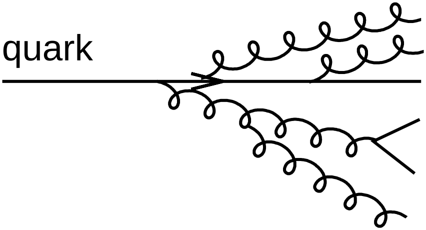
Gluon emission:

$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



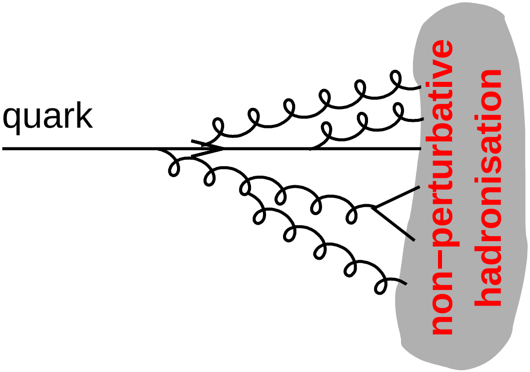
Gluon emission:

$$\int \alpha_s \frac{dE}{E}\frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



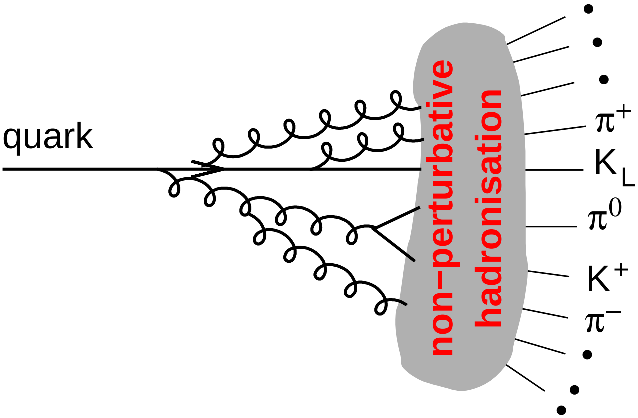
Gluon emission:

$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



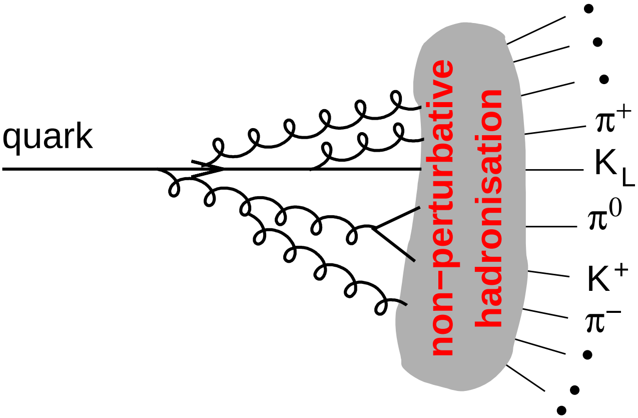
Gluon emission:

$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



Gluon emission:

$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

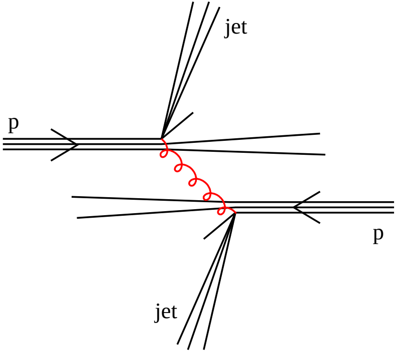
At low scales:

$$\alpha_s \to 1$$

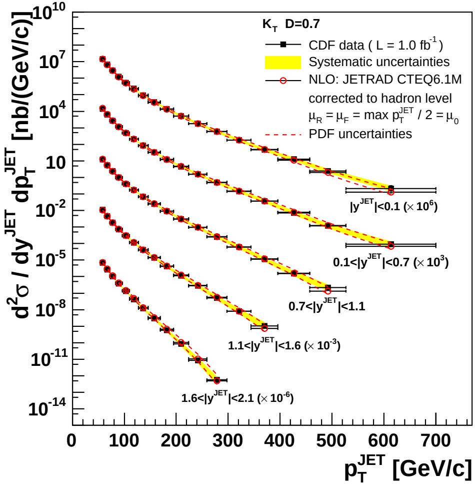
High-energy partons unavoidably lead to collimated bunches of hadrons

Jets from scattering of partons

Jets are unavoidable at hadron colliders, e.g. from parton scattering



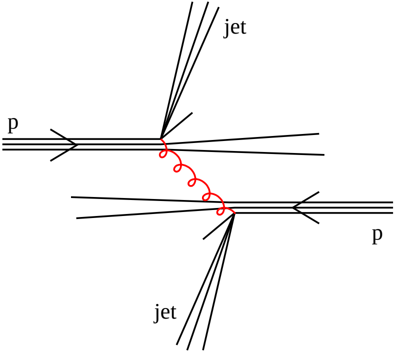
Tevatron results



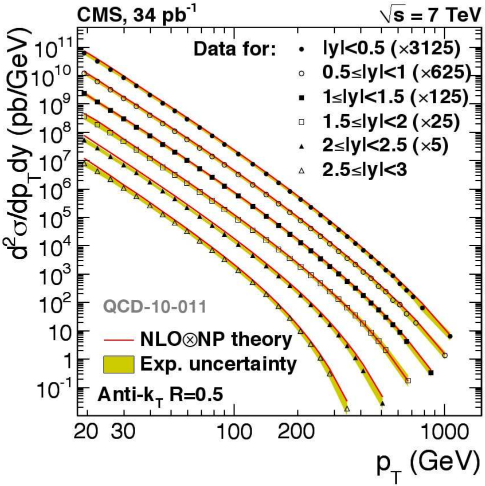
Jet cross section: data and theory agree over many orders of magnitude ⇔ probe of underlying interaction

Jets from scattering of partons

Jets are unavoidable at hadron colliders, e.g. from parton scattering



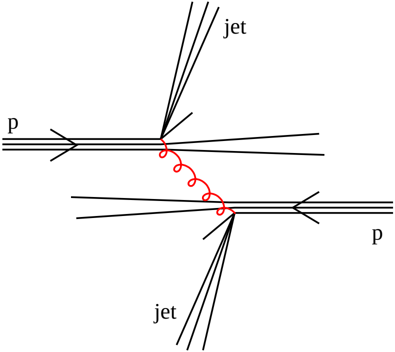
Latest CMS results!



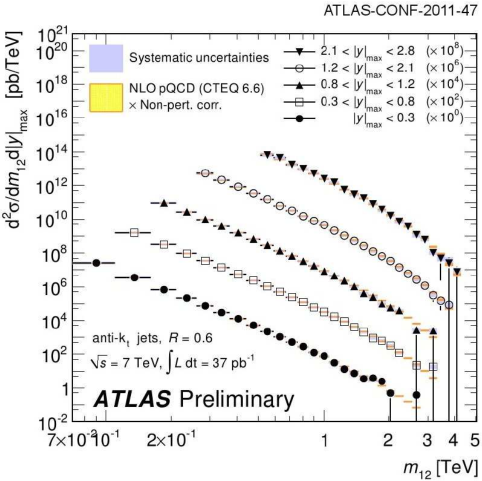
Jet cross section: data and theory agree over many orders of magnitude ⇔ probe of underlying interaction

Jets from scattering of partons

Jets are unavoidable at hadron colliders, e.g. from parton scattering

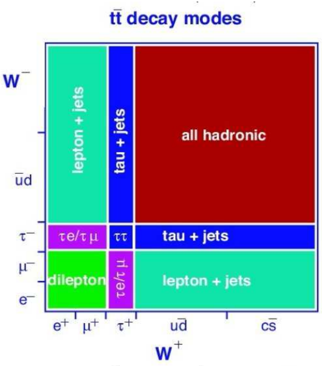


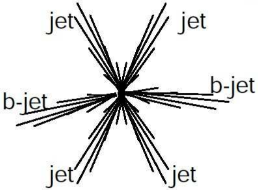
Latest ATLAS results!



Jet cross section: data and theory agree over many orders of magnitude ⇔ probe of underlying interaction

Jets from heavy decays





All-hadronic
(BR~46%, huge bckg)

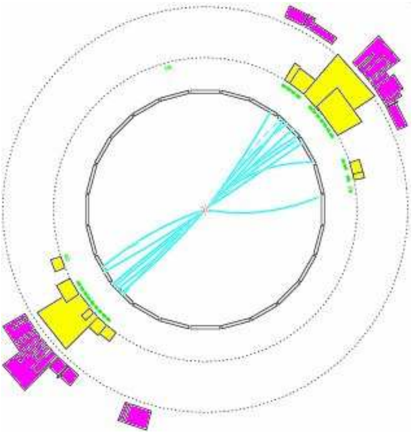
Heavy objects: multi-jet final-states

- 10^7 $t\bar{t}$ pairs for 1 fb^{-1} @ 14 TeV
- Vast # of QCD multijet events

# jets	# events for $1\,\mathrm{fb}^{-1}$
3	$2 \cdot 10^{10}$
4	$5 \cdot 10^{9}$
5	$1 \cdot 10^{9}$
6	$3 \cdot 10^{8}$
7	$1 \cdot 10^{8}$
8	$4 \cdot 10^{7}$

Tree level
$p_t(\text{jet}) > 20$ GeV, $\Delta R_{ij} > 0.4$, $|y_{ij}| < 2.5$
Gleisman & Höche '08

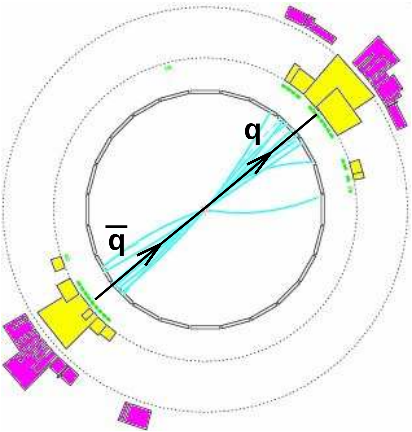
Seeing v. defining jets



Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

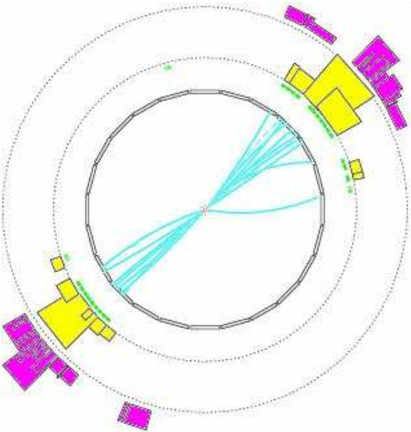
Seeing v. defining jets



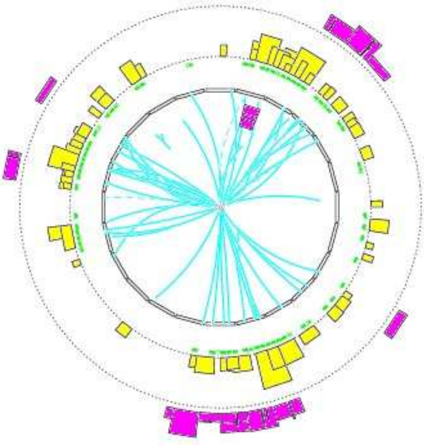
Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets



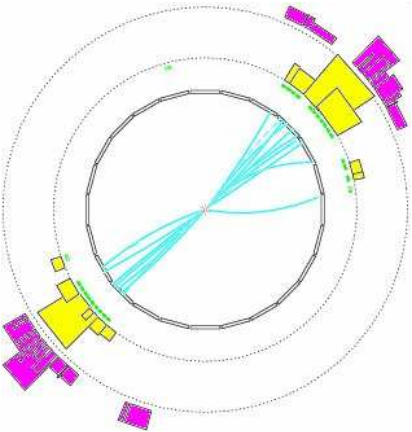
Jets are what we see.
Clearly(?) 2 jets here



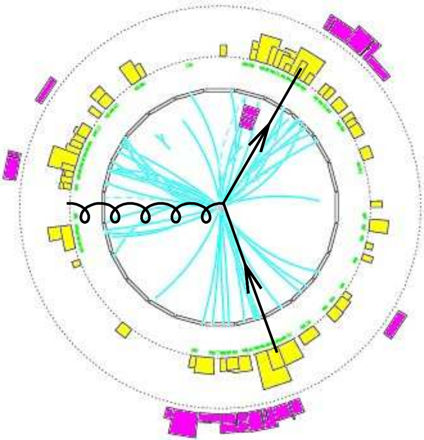
How many jets do you see?

Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets

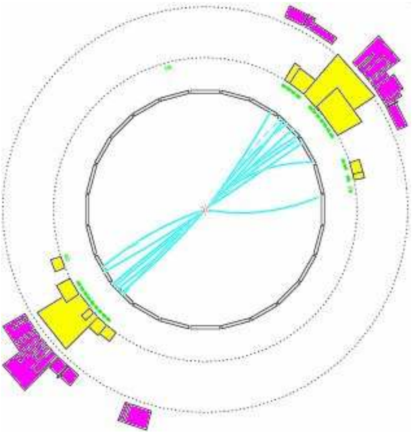


Jets are what we see.
Clearly(?) 2 jets here

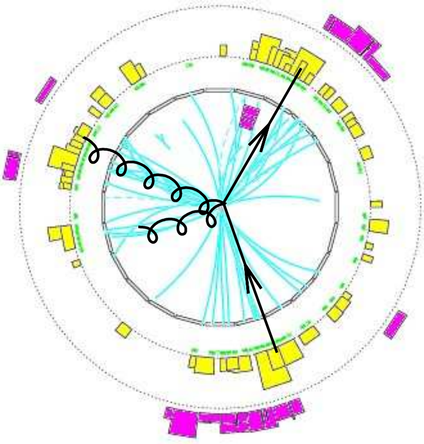


How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets

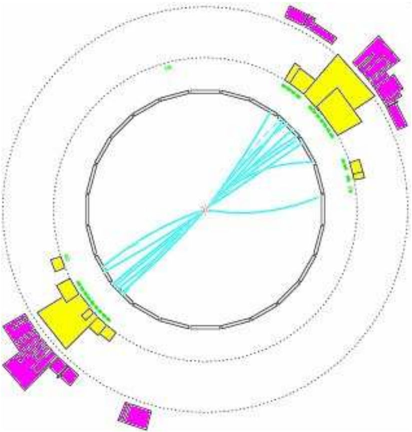


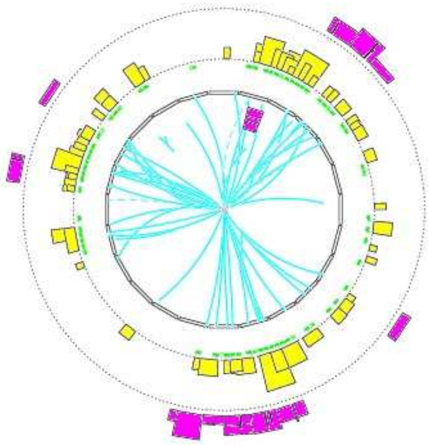
Jets are what we see.
Clearly(?) 2 jets here



How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets





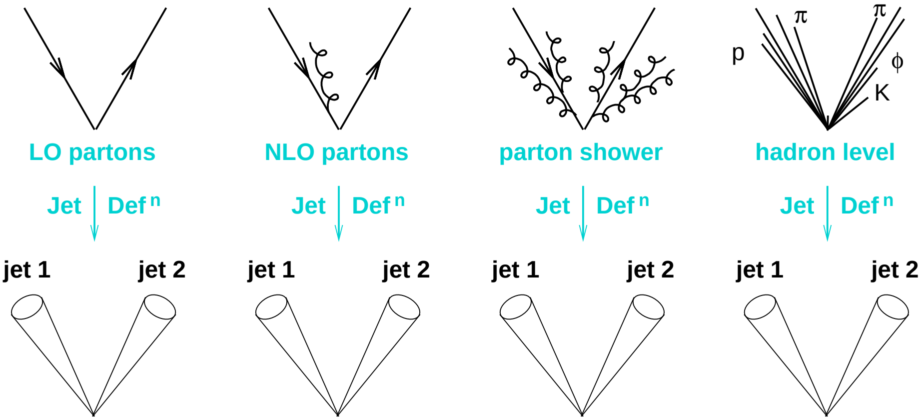
Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

Jet definitions

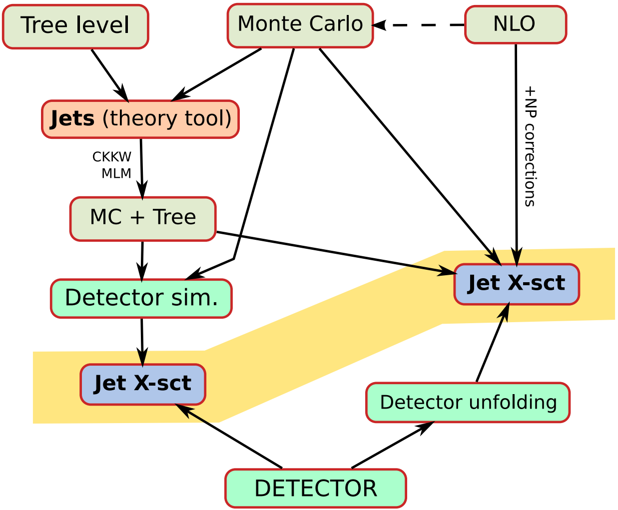
- *A jet definition is a fully specified set of rules for projecting information from 100's of hadrons, onto a handful of parton-like objects:*
  - or project 1000's of calorimeter towers
  - or project dozens of (showered) partons
  - or project a handful of (unshowered) partons
- Resulting objects (jets) used for many things, e.g. :
  - reconstructing decaying massive particles — e.g. top $\rightarrow$ 3 jets
  - constraining proton structure
  - as a theoretical tool to attribute structure to events — MLM/CKKW matching
- You *lose much information* in projecting event onto jet-like structure:
  - Sometimes information you had no idea how to use
  - Sometimes information you may not trust, or of no relevance

Jets as projections



Projection to jets should be resilient to QCD effects

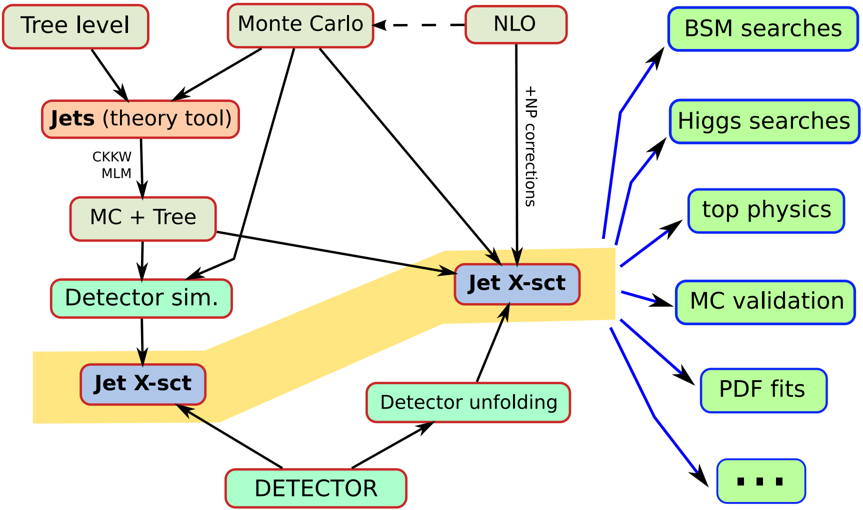
QCD jets flowchart



Jet (definitions) provide central link between expt., "theory" and theory

And jets are an input to almost all analyses

QCD jets flowchart



Jet (definitions) provide central link between expt., “theory” and theory

And jets are an input to almost all analyses

These lectures

Aims: to provide you with

- the "basics" needed to understand what goes into current jet-based measurements;
- some insight into the issues that are relevant when thinking about a jet measurement

Structure:

- General considerations
- Common jet definitions at LHC — Today
- Inside jets — Thursday
- Physics with jets — Friday

Defining jets

There is no unique jet definition

The construction of a jet is unavoidably ambiguous. On at least two fronts:

1. which particles get put together into a common jet? — Jet algorithm + parameters
2. how do you combine their momenta? — Recombination scheme

Most commonly used: direct 4-vector sums (E-scheme)

Taken together, these different elements specify a choice of jet definition

There is no unique jet definition

The construction of a jet is unavoidably ambiguous. On at least two fronts:

1. which particles get put together into a common jet? — Jet algorithm + parameters
2. how do you combine their momenta? — Recombination scheme

   Most commonly used: direct 4-vector sums (E-scheme)

Taken together, these different elements specify a choice of jet definition

The power of ambiguity

- Physical results (particle discovery, masses, PDFs, coupling) should be independent of your choice of jet definition

  a bit like renormalisation scale/scheme invariance

  Tests independence on modelling of radiation, hadronisation, etc.

- Except when there is a good reason for this not to be the case

Jetography, like photography

- Fine detail on boarding pass — shoot from close up, focus = 40cm

  [look for gate]

- Keep focus at 40cm
- Reset focus to 3m

  Catch correct plane

Jetography, like photography

- Fine detail on boarding pass — shoot from close up, focus = 40cm

  [look for gate]

- Keep focus at 40cm
- Reset focus to 3m

  Catch correct plane

Jetography, like photography

- Fine detail on boarding pass — shoot from close up, focus = 40cm

[look for gate]

- Keep focus at 40cm
- Reset focus to 3m

Catch correct plane

Jetography, like photography

- Fine detail on boarding pass — shoot from close up, focus = 40cm

  [look for gate]

- Keep focus at 40cm
- Reset focus to 3m

  Catch correct plane

Not all ambiguity is allowed

Jets should be **invariant** with respect to certain modifications of the event:

- collinear splitting
- infrared emission

Why?

- Because otherwise lose real-virtual cancellation in NLO/NNLO QCD calculations → divergent results
- Hadron-level 'jets' would become fundamentally non-perturbative
- Detectors can resolve neither full collinear nor full infrared event structure

Known as **infrared and collinear safety**

Not all ambiguity is allowed

Jets should be **invariant** with respect to certain modifications of the event:

- collinear splitting
- infrared emission

Why?

- Because otherwise lose real-virtual cancellation in NLO/NNLO QCD calculations $\rightarrow$ divergent results
- Hadron-level 'jets' would become fundamentally non-perturbative
- Detectors can resolve neither full collinear nor full infrared event structure

Known as **infrared and collinear safety**

Not all ambiguity is allowed

Jets should be **invariant** with respect to certain modifications of the event:

- collinear splitting
- infrared emission

Why?

- Because otherwise lose real-virtual cancellation in NLO/NNLO QCD calculations $\rightarrow$ divergent results
- Hadron-level 'jets' would become fundamentally non-perturbative
- Detectors can resolve neither full collinear nor full infrared event structure

Known as **infrared and collinear safety**

Two main classes of jet alg.

Sequential recombination (k_t, etc.)

- bottom-up
- successively undoes QCD branching

Cone

- top-down
- centred around idea of an 'invariant', directed energy flow

Cones: most widely used at Tevatron
Seq. rec.: most widely used at LHC and HERA

In this lecture we'll concentrate on the sequential recombination algorithms

Two main classes of jet alg.

Sequential recombination (k_t, etc.)

- bottom-up
- successively undoes QCD branching

Cone

- top-down
- centred around idea of an 'invariant', directed energy flow

Cones: most widely used at Tevatron
Seq. rec.: most widely used at LHC and HERA

In this lecture we'll concentrate on the sequential recombination algorithms

Sequential recombination jet algorithms

starting with a classic e^+e^- algorithm

Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

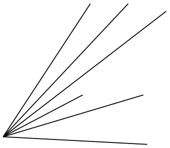
Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

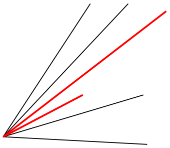


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

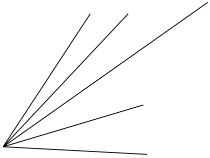


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

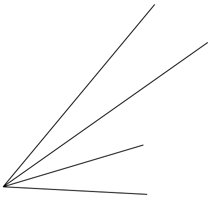


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

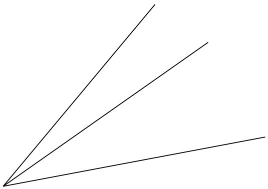


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

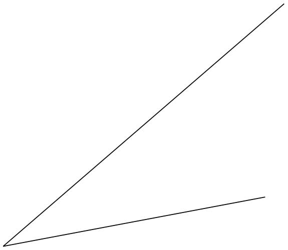


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.



The main questions are:

- How do you choose which pair of particles to combine at any given stage?
- When do you stop combining them?

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \simeq \frac{2\alpha_s C_A}{\pi} \frac{dE_j}{\min(E_i, E_j)} \frac{d\theta_{ij}}{\theta_{ij}}\,, \qquad (E_j \ll E_i\,,\ \theta_{ij} \ll 1)\,.$$

To invert branching process, take pair with strongest divergence between them — they're the most *likely* to belong together.

This is basis of **k_t/Durham algorithm** (e^+e^-):

1. Calculate (or update) distances between all particles i and j:

$$y_{ij} = \frac{2\min(E_i^2, E_j^2)(1-\cos\theta_{ij})}{Q^2}$$

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \simeq \frac{2\alpha_s C_A}{\pi}\frac{dE_j}{\min(E_i,E_j)}\frac{d\theta_{ij}}{\theta_{ij}}\,, \qquad (E_j \ll E_i\,,\;\; \theta_{ij} \ll 1)\,.$$

To invert branching process, take pair with strongest divergence between them — they're the most *likely* to belong together.

This is basis of **$\mathbf{k_t}$/Durham algorithm** (e^+e^-):

1. Calculate (or update) distances between all particles i and j:

$$y_{ij} = \frac{2\min(E_i^2, E_j^2)(1-\cos\theta_{ij})}{Q^2}$$

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

[dk ... 1) .

To inv ... een
them

The algorithm has one parameter

- y_{cut}: sets minimal relative transverse momentum between any pair of jets

This is basis of **$\mathbf{k_t}$/Durham algorithm** (e^+e^-):

1. Calculate (or update) distances between all particles i and j:

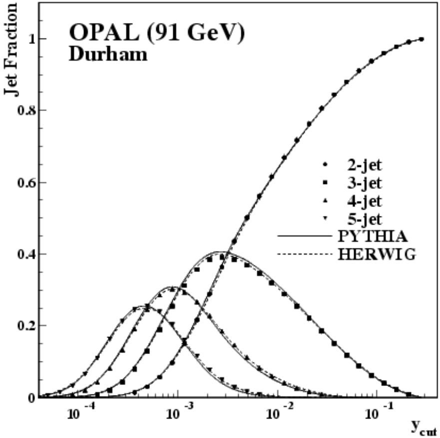
$$y_{ij} = \frac{2\min(E_i^2, E_j^2)(1-\cos\theta_{ij})}{Q^2}$$

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm features



- Gives hierarchy to event and jets
  Event can be charaterised by y_{23}, y_{34}, y_{45}.
- Resolution parameter related to minimal transverse momentum between jets

Most widely-used jet algorithm in e^+e^-

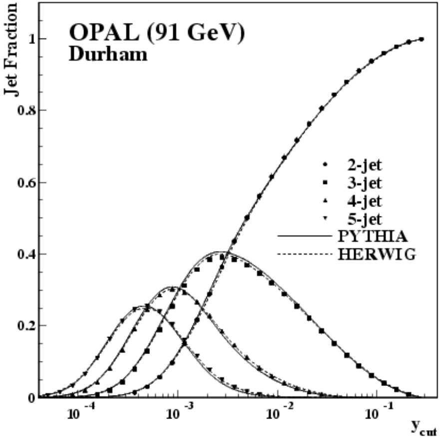
- Collinear safe: collinear particles recombined early on
- Infrared safe: soft particles have no impact on rest of clustering seq.

k_t/Durham algorithm features

- Gives hierarchy to event and jets

Event can be charaterised by y_{23}, y_{34}, y_{45}.

- Resolution parameter related to minimal transverse momentum between jets



Most widely-used jet algorithm in e^+e^-

- Collinear safe: collinear particles recombined early on
- Infrared safe: soft particles have no impact on rest of clustering seq.

k_t alg. at hadron colliders

1st attempt

- Lose absolute normalisation scale Q. So use unnormalised d_{ij} rather than y_{ij}:

$$d_{ij} = 2\min(E_i^2, E_j^2)(1 - \cos\theta_{ij})$$

- Now also have *beam remnants* (go down beam-pipe, not measured)
  Account for this with particle-beam distance

$$d_{iB} = 2E_i^2(1 - \cos\theta_{iB})$$

squared transv. mom. wrt beam

k_t alg. at hadron colliders

2nd attempt: make it longitudinally boost-invariant

- Formulate in terms of rapidity (y), azimuth (ϕ), p_t

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\Delta R_{ij}^2\,, \qquad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

NB: not η_i, E_{ti}

- Beam distance becomes

$$d_{iB} = p_{ti}^2$$

squared transv. mom. wrt beam

Apart from measures, just like e^+e^- alg.
Known as **exclusive k_t algorithm**.

Problem: at hadron collider, no single fixed scale (as in Q in e^+e^-). So how do you choose d_{cut}? See e.g. Seymour & Tevlin '06

k_t alg. at hadron colliders

3rd attempt: **inclusive k_t algorithm**

- Introduce angular radius R (NB: dimensionless!)

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}\,, \qquad d_{iB} = p_{ti}^2$$

- 1. Find smallest of d_{ij}, d_{iB}
  2. if ij, recombine them
  3. if iB, call i a jet and remove from list of particles
  4. repeat from step 1 until no particles left.

S.D. Ellis & Soper, '93; the simplest to use

Jets all separated by at least R on y, ϕ cylinder.

NB: number of jets not IR safe (soft jets near beam); number of jets above p_t cut **is** IR safe.

k_t alg. at hadron colliders

3rd attempt: **inclusive k_t algorithm**

- Introduce angular radius R (NB: dimensionless!)

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = p_{ti}^2$$

- 1. F
  2. if
  3. if
  4. re

to use

Jets a

Two parameters to remember

- **R**: sets $y-\phi$ reach of the jet; minimal interjet separation
- **p_t cut** on the jets

These parameters are common to all widely used hadron-collider jet algorithms.

NB: number of jets not IR safe (soft jets near beam); number of jets above p_t cut **is** IR safe.

k_t is a form of Hierarchical Clustering

Fast Hierarchical Clustering and Other Applications of Dynamic Closest Pairs

David Eppstein
UC Irvine

We develop data structures for dynamic closest pair problems with arbitrary distance functions, that do not necessarily come from any geometric structure on the objects. Based on a technique previously used by the author for Euclidean closest pairs, we show how to insert and delete objects from an n-object set, maintaining the closest pair, in $O(n \log^2 n)$ time per update and $O(n)$ space. With quadratic space, we can instead use a quadtree-like structure to achieve an optimal time bound, $O(n)$ per update. We apply these data structures to hierarchical clustering, greedy matching, and TSP heuristics, and discuss other potential applications in machine learning, Gröbner bases, and local improvement algorithms for partition and placement problems. Experiments show our new methods to be faster in practice than previously used heuristics.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms**]: Nonnumeric Algorithms

General Terms: Closest Pair, Agglomerative Clustering

Additional Key Words and Phrases: TSP, matching, conga line data structure, quadtree, nearest neighbor heuristic

1. INTRODUCTION

Hierarchical clustering has long been a mainstay of statistical analysis, and clustering based methods have attracted attention in other fields: computational biology (reconstruction of evolutionary trees; tree-based multiple sequence alignment), scientific simulation (n-body problems), theoretical computer science (network design and nearest neighbor searching) and of course the web (hierarchical indices such as Yahoo). Many clustering methods have been devised and used in these applications, but less effort has gone into algorithmic speedups of these methods.

In this paper we identify and demonstrate speedups for a key subroutine used in several clustering algorithms, that of maintaining closest pairs in a dynamic set of objects. We also describe several other applications or potential applications of the

Idea behind k_t alg. is to be found over and over in many areas of (computer) science.

k_t in action

$\mathbf{k_t}$ alg.: Find smallest of

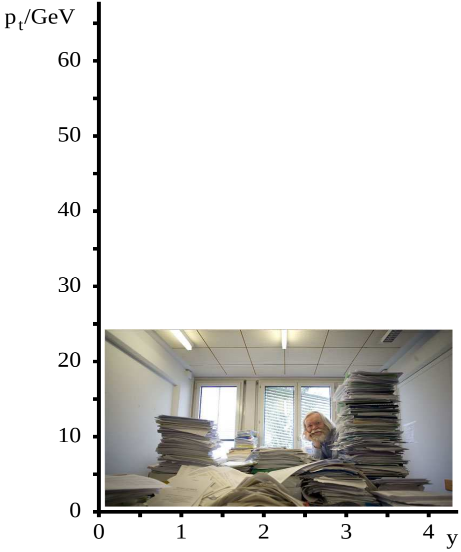
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

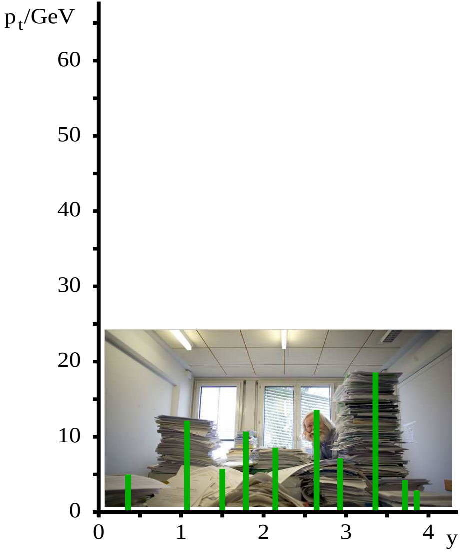
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

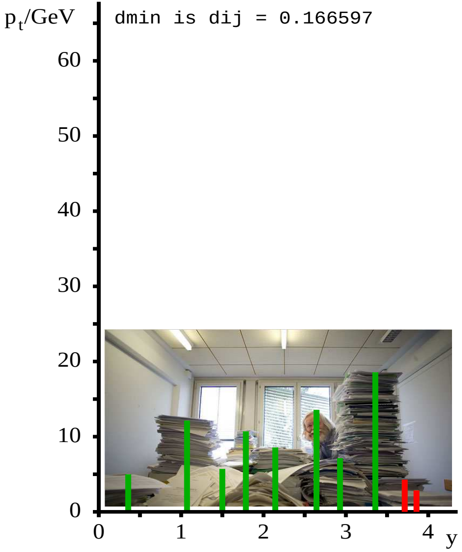
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

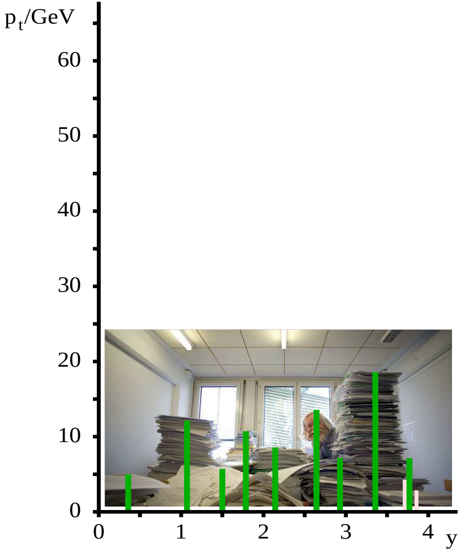
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

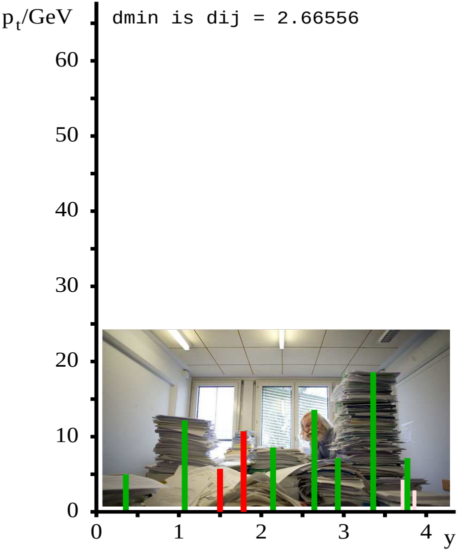
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

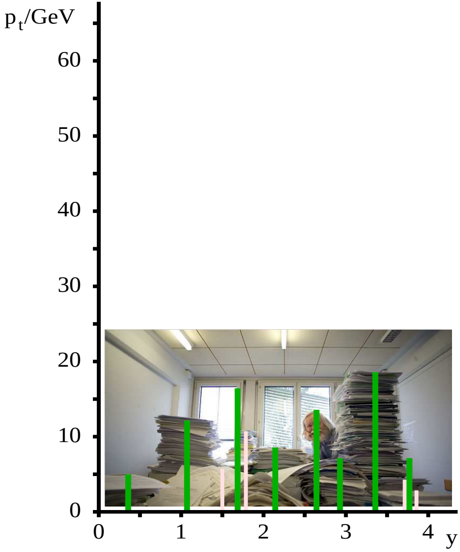
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

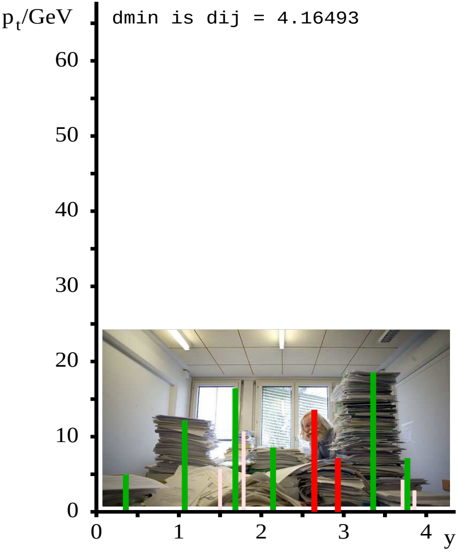
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

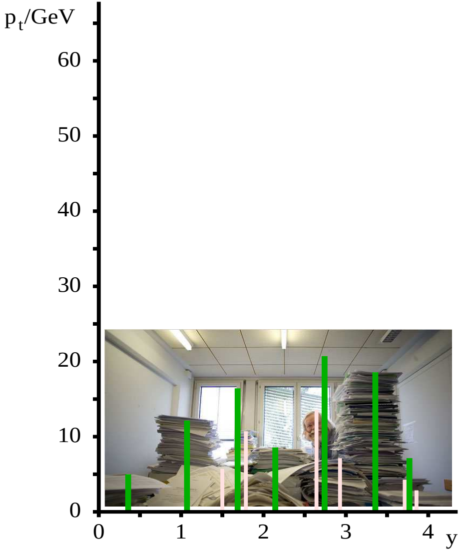
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

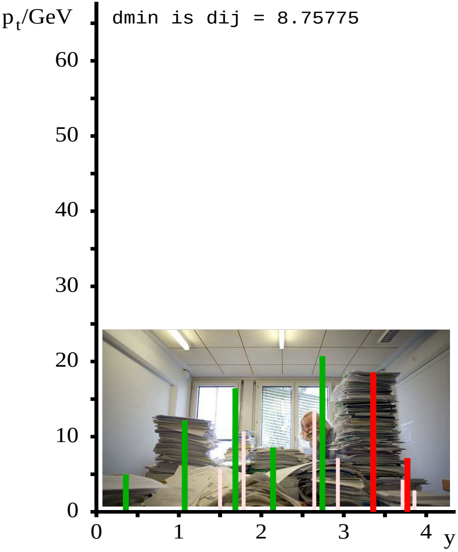
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

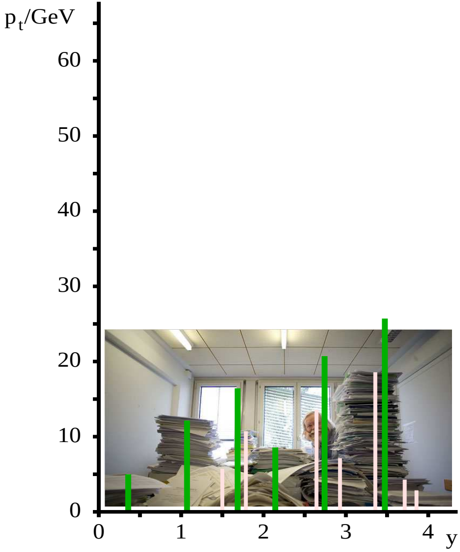
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

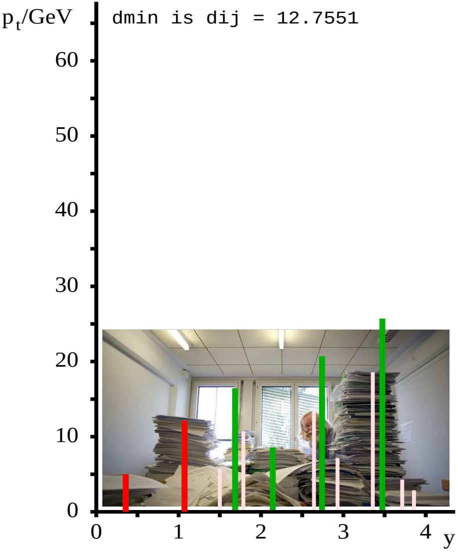
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

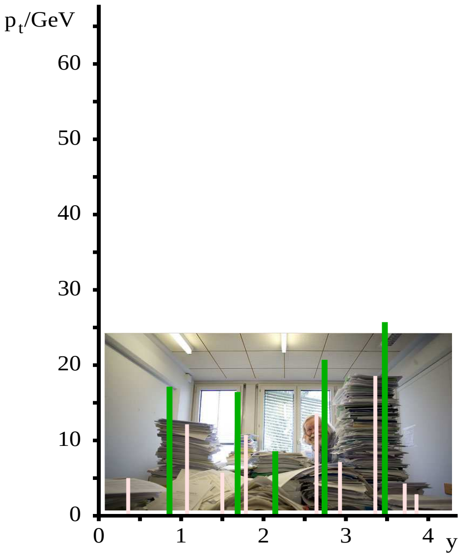
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

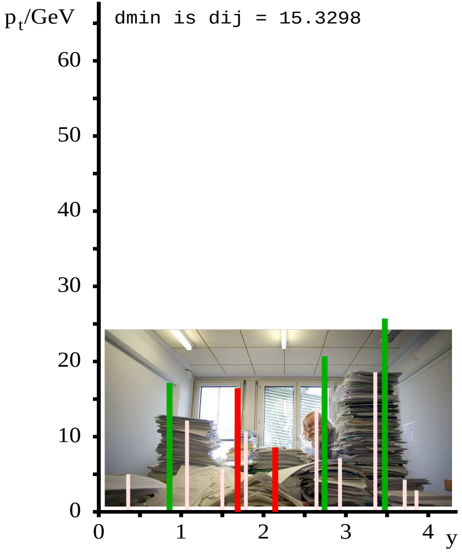
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

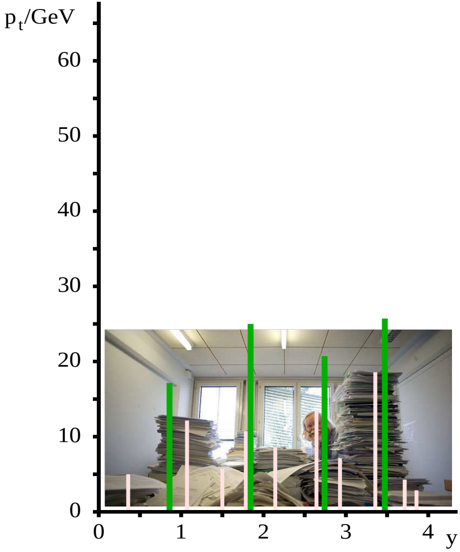
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

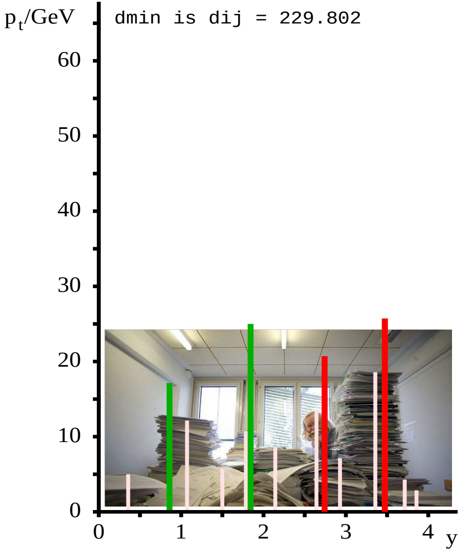
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

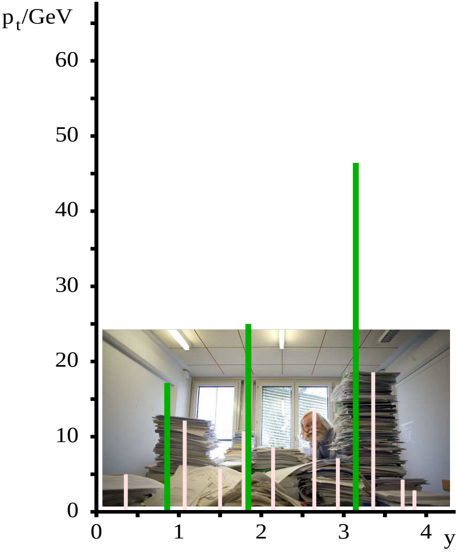
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

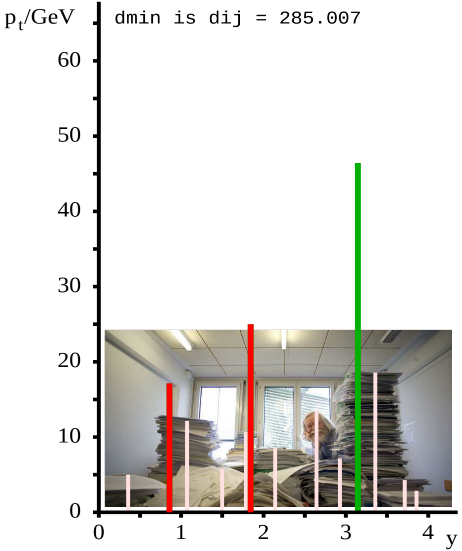
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

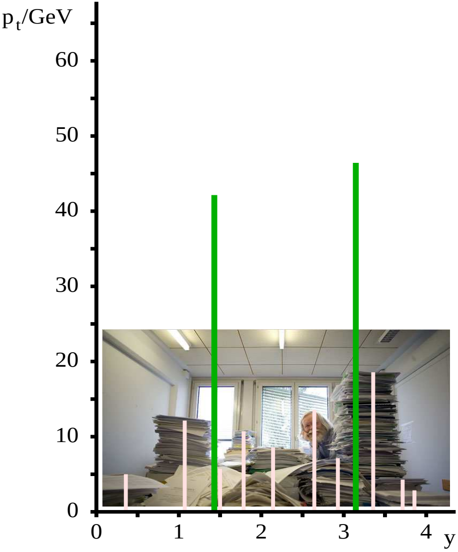
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

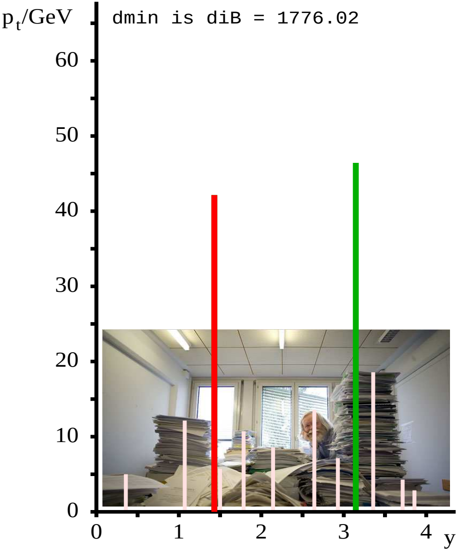
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

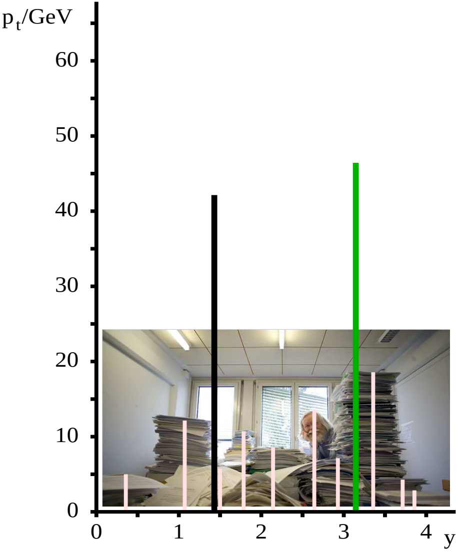
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

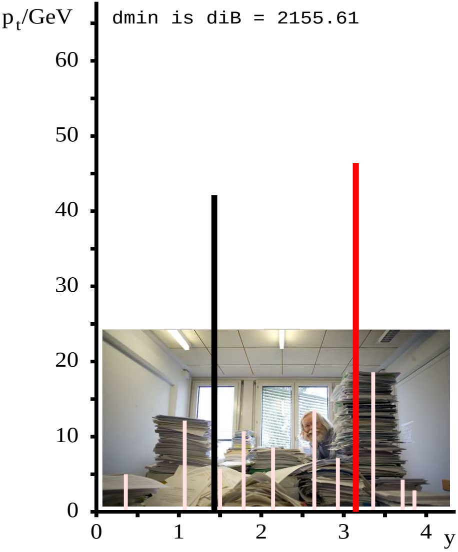
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



$\mathbf{k_t}$ alg.: Find smallest of

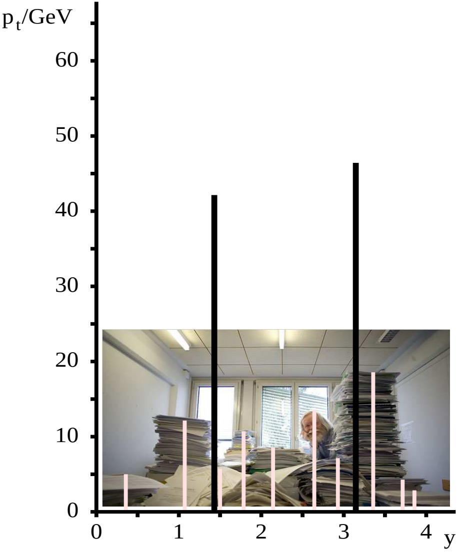
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

- If d_{ij} recombine
- if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

The k_t algorithms form one of several "families" of sequential recombination jet algorithm

Others differ in:

1. the choice distance measure between pairs of particles
[i.e. the relative priority given to soft and collinear divergences]

2. using $3 \to 2$ clustering rather than $2 \to 1$
[ARCLUS; not used at hadron colliders, so won't discuss it more]

Sequential recombination variants

Cambridge/Aachen: *the simplest of hadron-collider algorithms*

- Recombine pair of objects closest in ΔR_{ij}
- Repeat until all $\Delta R_{ij} > R$ — remaining objects are jets

Dokshitzer, Leder, Moretti, Webber '97 (Cambridge): more involved e^+e^- form
Wobisch & Wengler '99 (Aachen): simple inclusive hadron-collider form

C/A privileges the collinear divergence of QCD;
it 'ignores' the soft one

Anti-k_t

Anti-k_t: *formulated similarly to k_t, but with*

$$d_{ij} = \min\left(\frac{1}{k_{ti}^2}, \frac{1}{k_{tj}^2}\right)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{k_{ti}^2}$$

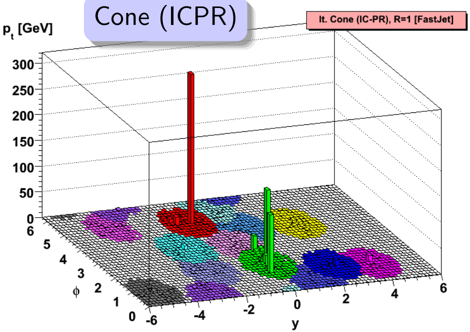
Cacciari, GPS & Soyez, '08 [+ Delsart unpublished]

Anti-k_t privileges the collinear divergence of QCD and disfavours clustering between pairs of soft particles

Most pairwise clusterings involve at least one hard particle

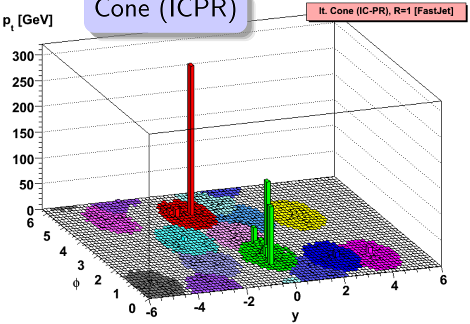
Essential characteristic of cones?

Cone (ICPR)



Essential characteristic of cones?

Cone (ICPR)

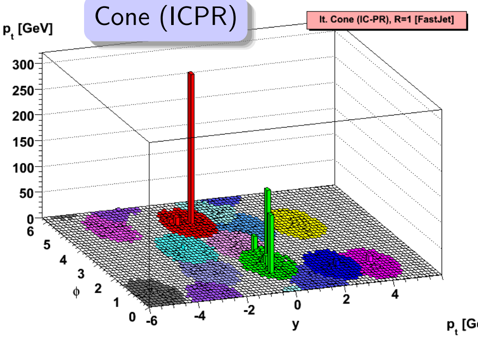


(Some) cone algorithms give **circular** jets in $y - \phi$ plane

Much appreciated by experiments e.g. for acceptance corrections

Essential characteristic of cones?

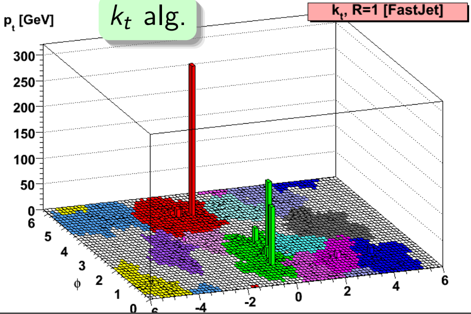
Cone (ICPR)



(Some) cone algorithms give **circular** jets in $y - \phi$ plane

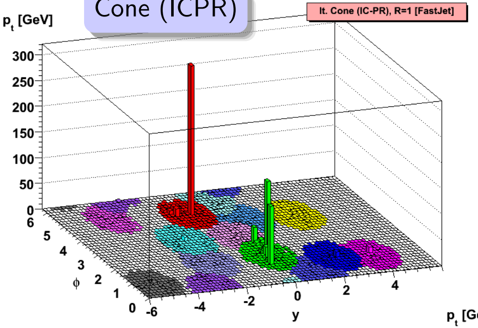
Much appreciated by experiments — e.g. for acceptance corrections

k_t alg.



Essential characteristic of cones?

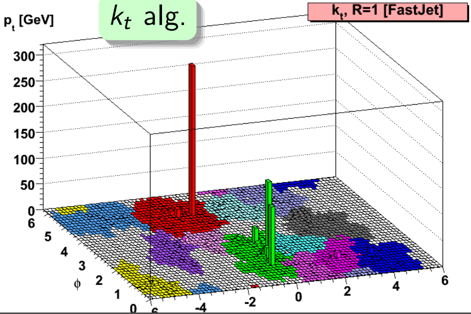
Cone (ICPR)



(Some) cone algorithms give **circular** jets in $y - \phi$ plane

Much appreciated by experiments e.g. for acceptance corrections

k_t alg.



k_t jets are **irregular**

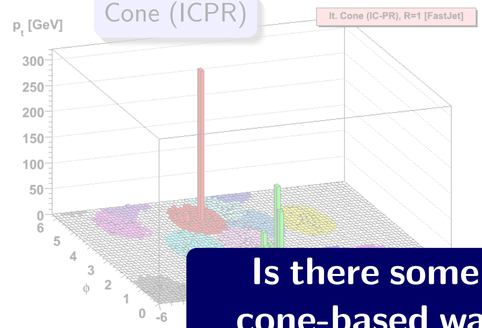
Because soft junk clusters together first:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2$$

Regularly held against k_t

Essential characteristic of cones?

Cone (ICPR)



(Some) cone algorithms give **circular** jets in $y - \phi$ plane

Much appreciated by experiments e.g. for acceptance corrections

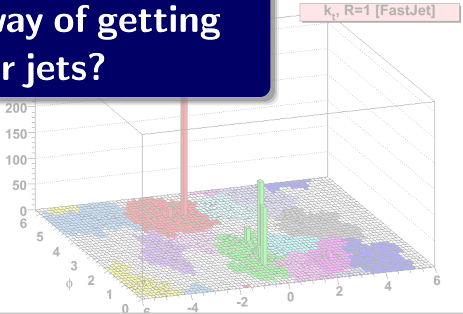
Is there some other, non cone-based way of getting circular jets?

k_t jets are **irregular**

Because soft junk clusters together first:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2$$

Regularly held against k_t



Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\text{: } d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\text{: } d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

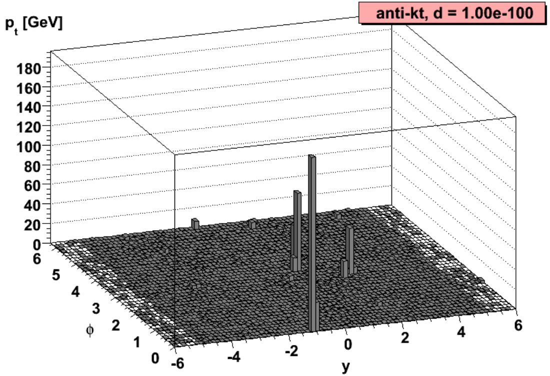
Hard stuff clusters with nearest neighbour

Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

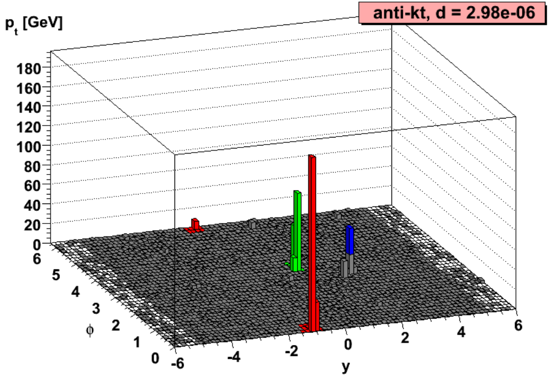


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

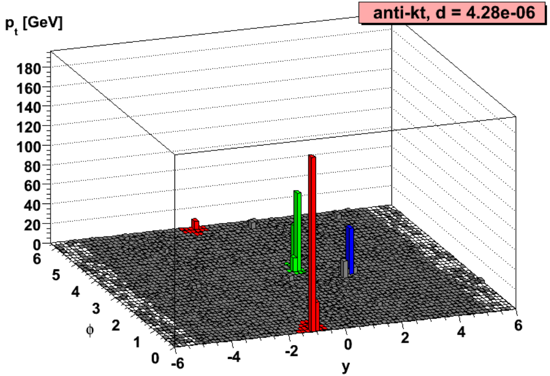


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

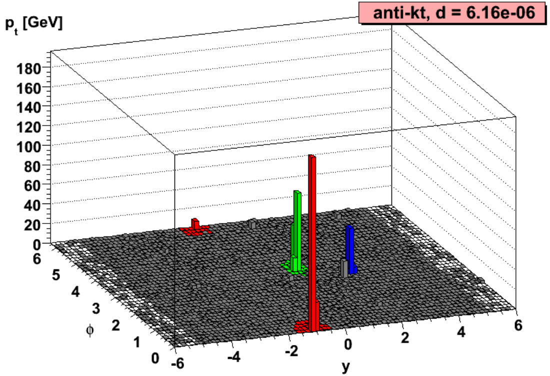


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

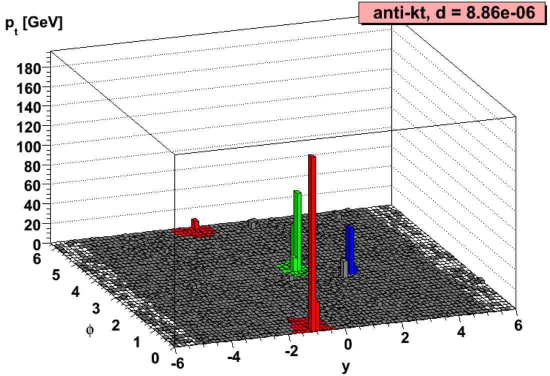


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

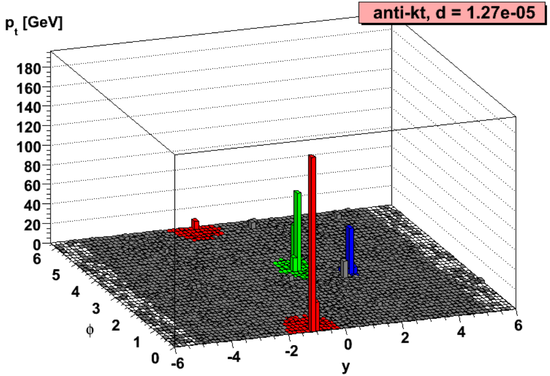


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

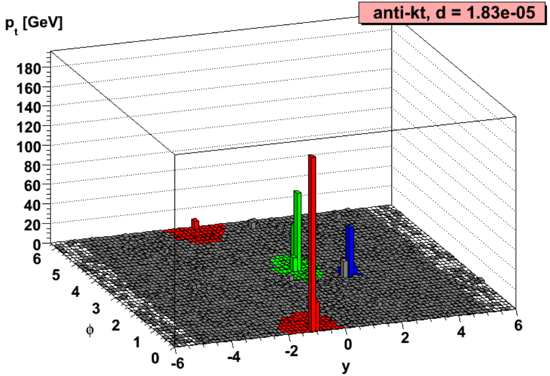


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\textbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

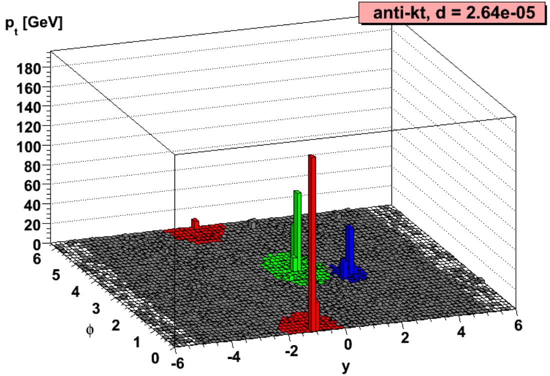


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

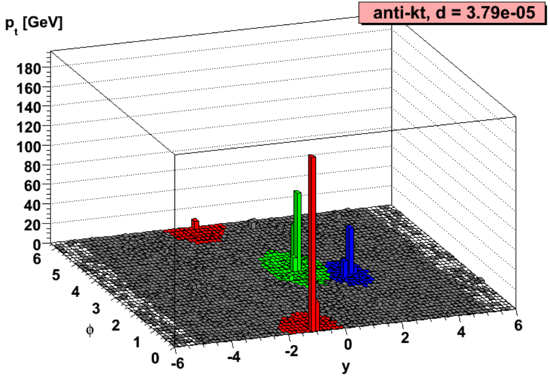


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\textbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

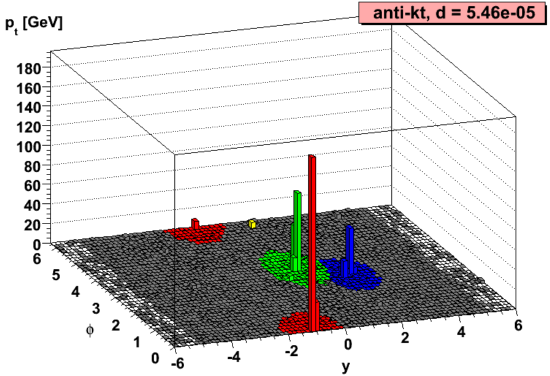


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

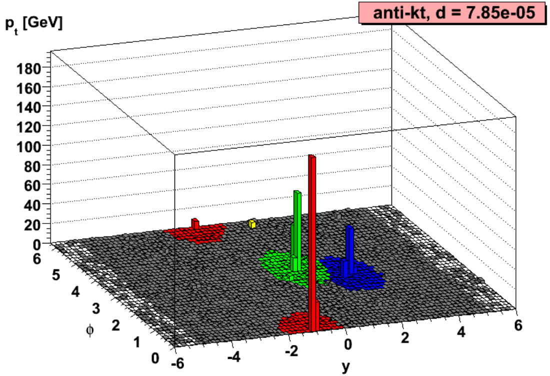


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

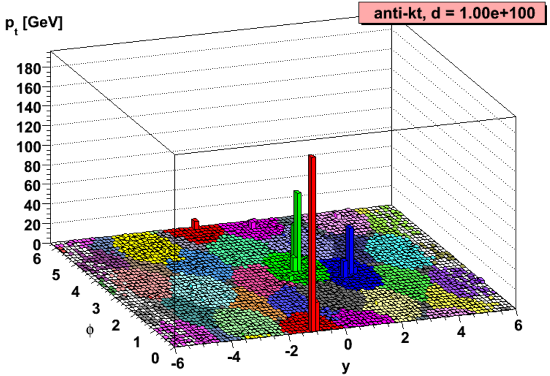


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

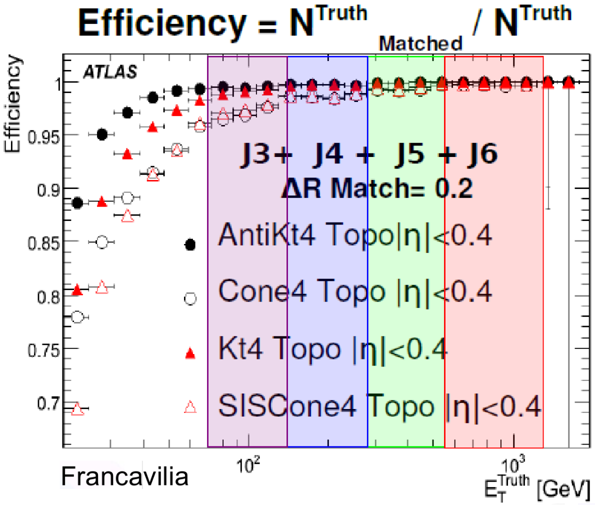
$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\textbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour



anti-k_t gives cone-like jets without using stable cones

Anti-k_t experimental performance

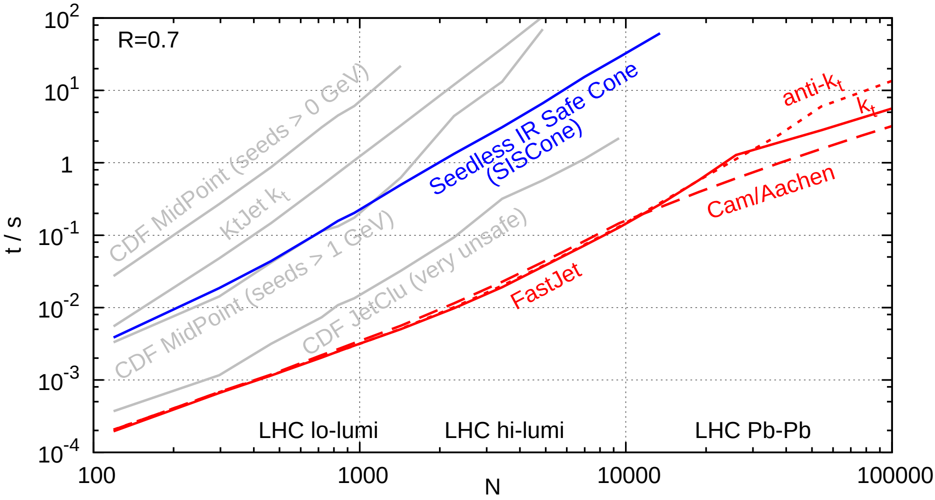


As good as, or better than all previous experimentally-favoured algorithms.

Essentially because anti-k_t has linear response to soft particles.

And it's also infrared and collinear safe (needed for theory calcs.)

FastJet: time to cluster N particles



Summary

Today we've examined why we need jets and looked at some of the logic behind the way they're defined.

Of the different algorithms we've discussed, the one that's most widely used at LHC today is anti-k_t.

But the other algorithms we've seen will also play a role in the forthcoming lectures (and at LHC!).

Tomorrow's subject will be *the internal structure of jets*.