QCD (for LHC)
Lecture 4

1. Merging parton showers and fixed order
2. Jets

Gavin Salam

LPTHE, CNRS and UPMC (Univ. Paris 6)

At the 2009 European School of High-Energy Physics
June 2009, Bautzen, Germany
Tree-level (LO) gives decent description of multi-jet structure

NLO gives good normalisation

Parton-shower gives good behaviour in soft-collinear regions and fully exclusive final state.

*Can we combine the advantages of all three?*
Difficulties in merging Tree-level(s) + PS?

Suppose you ask for $Z+\text{jet}$ as your initial hard process in Pythia/Herwig.

- They contain the correct ME for $Z+j$.
- But you want $Z+2\text{j}$ to be correct too.

Naive approach: you could also generate $Z+2\text{j}$ events with Alpgen (or Madgraph, etc.) and run the shower from those configurations too.
Add $Z+1\text{jet}$, $Z+2\text{jet} + \text{shower}$
Add $Z+1\text{jet}$, $Z+2\text{jet} + \text{shower}$
Add $Z+1\text{jet}, Z+2\text{jet} + \text{shower}$

- Combining PS + FO
- Tree-level + PS

shower $Z+\text{parton}$

$Z+2\text{partons}$
Add $Z+1\text{jet}$, $Z+2\text{jet} + \text{shower}$

shower $Z+\text{parton}$  +  shower $Z+2\text{partons}$
Add $Z+1\text{jet}, \ Z+2\text{jet} \ + \ \text{shower}

\[
\begin{align*}
\text{shower} \ Z+\text{parton} & \quad + \quad \text{shower} \ Z+2\text{partons} \\
& \quad \quad \text{v.} \quad \quad \text{shower of } Z+\text{parton}
\end{align*}
\]

\text{generates hard gluon}
Add $Z+1\text{jet}$, $Z+2\text{jet} + \text{shower}$

- shower $Z+\text{parton}$
- shower $Z+2\text{partons}$
- shower of $Z+\text{parton}$ generates hard gluon
Combining PS + FO

Tree-level + PS

Add Z+1jet, Z+2jet + shower

DOUBLE COUNTING

shower Z+parton

+  

shower Z+2partons

v.

shower of Z+parton generates hard gluon

Double counting + associated issues with virtual corrections are the main problems when merging PS + ME
ME + PS merging is an attempt to solve this. There are many variants. One common one is “MLM matching” — a summary of it is:

- Introduce a cutoff $Q_{ME}$
- Use the matrix elements to generate tree-level events for $Z+1$ parton, $Z+2$ partons, ... $Z+N$ partons, where all partons must have $p_t > Q_{ME}$, and are separated from the others by some angle $R_{ME}$.

  Numbers of events are in proportion to their cross sections with these cuts

- Take one of these tree level events, say with $n$-partons.
- Shower it with your favourite Parton Shower program.
- Identify all jets that have $p_t > Q_{\text{merge}}$ (chosen $\gtrsim Q_{ME}$)
- If each parton corresponds to one of the jets ($\equiv$ is nearby in angle) and there are no extra jets above scale $Q_{\text{merge}}$, accept the event.
- Otherwise reject it.

[Replace $Q_{\text{merge}} \rightarrow p_{tn}$ if $n = N$]

NB: MLM stands for Michelangelo L. Mangano
Combining PS + FO

Tree-level + PS

MLM example

- Hard jets above scale $Q_{merge}$ have distributions given by tree-level ME
- Rejection procedure eliminates “double-counted” jets from parton shower
- Rejection generates Sudakov form factors between individual jet scales

How well? Depends on details of PS. One of the weaker points of MLM
Combining PS + FO

Tree-level + PS

- **ACCEP**
- **REJECT**

- $Q_{merge}$
- $p_t$ cut

**shower** $Z+$parton

**shower** $Z+2$partons

**shower** of $Z+$parton generates hard gluon

- Hard jets above scale $Q_{merge}$ have distributions given by tree-level ME
- Rejection procedure eliminates “double-counted” jets from parton shower
- Rejection generates Sudakov form factors between individual jet scales

How well? Depends on details of PS. One of the weaker points of MLM
QCD lecture 4 (p. 6)

Combining PS + FO

Tree-level + PS

MLM example

- **Q**
  - **merge**
  - **p**
  - **t cut**

> shower \( Z + \text{parton} \)

Hard jets above scale \( Q_{\text{merge}} \) have distributions given by tree-level ME

Rejection procedure eliminates “double-counted” jets from parton shower

Rejection generates Sudakov form factors between individual jet scales

How well? Depends on details of PS. One of the weaker points of MLM
MLM is the standard merging available from Alpgen

There are several other merging procedures on the market

- MLM à la MadGraph
  Mainly changes details of jet finding
- CKKW
  e.g. in Sherpa
- CKKW-L
  e.g. in Ariadne
- Pseudo Shower
  by Mrenna

They vary essentially in whether/how they match partons & jets, the definitions of the jets, and some include analytic Sudakov form factors (e.g. CKKW).

They all involve some implicit form of $p_t$ cutoff.

Usually physics well above cutoff is independent of cutoff?
Combining PS + FO
Tree-level + PS

$Z + 1$ jet

D0 Run II, $L=1.04$ fb$^{-1}$
Data at particle level
MCFM NLO

$\frac{1}{\sigma_{Z\gamma^*}} \times \frac{d\sigma_{Z\gamma^*}}{dp_T^{(1st\ jet)}}$ [1/GeV]

$Z/\gamma^* (\rightarrow ee) + 1$ jet + X
$65 < M_{ee} < 115$ GeV
Incl. in $p_T^e / y^e$
$R_{jet}^{cone} = 0.5, |y^{jet}| < 2.5$

Ratio to MCFM NLO

Data
MCFM NLO
MCFM LO
Scale unc.

Ratio to MCFM NLO

Data
PYTHIA S0
HERWIG+JIMMY
Scale unc.

Ratio to MCFM NLO

Data
ALPGEN+PYTHIA
SHERPA
Scale unc.

Ratio to MCFM NLO

Data
PYTHIA QW
Scale unc.

Ratio to MCFM NLO

Data
SHERPA
Scale unc.
Combining PS + FO

Tree-level + PS

Z + 2 jets

D0 Run II, L=1.04 fb⁻¹

\[
\frac{1}{\sigma_{Z/\gamma^*}} \times \frac{d^2 \sigma}{d p_T^2 d \eta} = 10^{-3} \text{ [GeV]}
\]

\(Z/\gamma^* (\rightarrow ee) + 2 \text{ jets} + X\)

65 < \(M_{ee} < 115 \text{ GeV}\)

Incl. in \(p_T^e/\gamma^*\)

\(R_{\text{cone}} = 0.5, |y^{\text{jet}}| < 2.5\)

Ratio to MCFM NLO

\(p_T (2^{\text{nd}} \text{ jet}) [\text{GeV}]\)

Data at particle level

MCFM NLO

(a)

(b)

(c)

(d)
ME + PS merging helps get correct $p_t$ dependence

It works much better than plain parton showers

Normalisation is still quite uncertain
Combining PS + FO
Tree-level + PS

Z + 3 jets

D0 Run II, $L=1.04$ fb$^{-1}$
Data at particle level
MCFM LO

(a) $\frac{d \sigma_{Z/\gamma^* (\rightarrow ee)}(3 \text{ jets} + X)}{d p_T(\text{3rd jet})}$

$65 < M_{ee} < 115$ GeV
Incl. in $p_T^e / y^e$
$R_{\text{cone}} = 0.5$, $|y^\text{jet}| < 2.5$

Ratio to MCFM LO

(b) Scale unc.

(c) Data
PYTHIA S0
HERWIG+JIMMY
Scale unc.

(d) Data
ALPGEN+PYTHIA
SHERPA
Scale unc.
Can we get parton-shower structure, with NLO accuracy (e.g. control of normalisation, pattern of radiation of extra parton)?
**MC@NLO ideas**

- Expand your Monte Carlo branching to first order in $\alpha_s$
  
  Rather non-trivial – requires deep understanding of MC

- Calculate differences wrt true $\mathcal{O}(\alpha_s)$ both in real and virtual pieces

- If your Monte Carlo gives correct soft and/or collinear limits, those differences are **finite**

- Generate extra partonic configurations with phase-space distributions proportional to those differences and shower them
Let’s imagine a problem with one phase-space dimension, e.g. $E$. Expand Monte Carlo cross section for emission with energy $E$:

$$\sigma^{\text{MC}} \equiv 1 \times \delta(E) + \alpha_s \sigma_{1R}^{\text{MC}}(E) + \alpha_s \sigma_{1V}^{\text{MC}} \delta(E) + \mathcal{O}(\alpha_s^2)$$

With true NLO real/virtual terms as $\alpha_s \sigma_{1R}(E)$ and $\alpha_s \sigma_{1V} \delta(E)$, define

$$\text{MC@NLO} = \text{MC} \times \left( 1 + \alpha_s (\sigma_{1V} - \sigma_{1V}^{\text{MC}}) + \alpha_s \int dE (\sigma_{1R}(E) - \sigma_{1R}^{\text{MC}}(E)) \right)$$

All weights finite, but can be $\pm 1$

Processes include Frixione, Laenen, Motylinski, Nason, Webber, White ’02–’08

Higgs boson, single vector boson, vector boson pair, heavy quark pair, single top (with and without associated $W$), lepton pair and associated Higgs+$W/Z$
Aims to work around MC@NLO limitations

- the (small fraction of) negative weights
- the tight interconnection with a specific MC

Principle

- Write a simplified Monte Carlo that generates just one emission (the hardest one) which alone gives the correct NLO result.

  Essentially uses special Sudakov

  \[ \Delta(k_t) = \exp(-\int \text{exact real-radiation probability above } k_t) \]

- Lets your default parton-shower do branchings below that \( k_t \).

Processes include

\( pp \rightarrow \) Heavy-quark pair, Higgs, single vector-boson

\( pp \rightarrow W', e^+ e^- \rightarrow t\bar{t} \)
MC@NLO e.g.: $t\bar{t}$ $p_t$ distribution for LHC

- MC@NLO gets right normalisation
- correct behaviour at low $p_t$ ($\sim$ rescaled Herwig)
- correct behaviour at high $p_t$ ($\sim$ NLO)

figure from talk by Frixione ’04
You can merge many different tree-levels \((Z+1, Z+2, Z+3, \ldots)\) with parton showering together into a consistent sample.

Shapes should be OK, normalisation is rather uncertain

Procedures are flexible and general — but not necessarily the final word

You can merge NLO accuracy with parton showers for simple processes (at most one light jet — single top case)

Two main methods: MC@NLO / POWHEG

It is hard theory work — must be done on a case by case basis

Incorporation of different multiplicities \((Z+1, Z+2, Z+3, \ldots)\) consistently at NLO for each multiplicity, together with parton showering, is a current research problem.
We’ve completed our tour of predictive methods in collider QCD (LO, NLO, NNLO; parton showers; mergings and matchings)

The last topic of these lectures is jets
They’ve already arisen in various contexts; now look at them in detail
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?
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?
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?
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?
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?
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?
Jets as projections

Projection to jets provides “universal” view of event
Jet (definitions) provide central link between expt., "theory" and theory

And jets are an input to almost all analyses
Jet (definitions) provide central link between expt., “theory” and theory
And jets are an input to almost all analyses
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, e.g. jet angular radius $R$

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.

cf. Les Houches ’07 nomenclature accord

Ambiguity complicates life, but gives flexibility in one’s view of events → Jets non-trivial!
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? \textit{Jet algorithm} + parameters, e.g. jet angular radius $R$

2. how do you combine their momenta? \textit{Recombination scheme}
   Most commonly used: direct 4-vector sums ($E$-scheme)

\textbf{Taken together, these different elements specify a choice of jet definition}

Ambiguity complicates life, but gives flexibility in one’s view of events → Jets non-trivial!
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? \text{Jet algorithm + parameters, e.g. jet angular radius } R

2. how do you combine their momenta? \text{Recombination scheme}

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

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

\text{cf. Les Houches '07 nomenclature accord}

\textbf{Ambiguity complicates life, but gives flexibility in one’s view of events} → \textbf{Jets non-trivial!}
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
Majority of QCD branching is soft & collinear, with following divergences:

\[ [dk_j] | M_{g\rightarrow g_i g_j}(k_j) | \sim \frac{2\alpha_s C_A}{\pi} \frac{dE_j}{\min(E_i, E_j)} \frac{d\theta_{ij}}{\theta_{ij}} , \quad (E_j \ll E_i, \ \theta_{ij} \ll 1). \]

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*:

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

2. Find smallest of *y*<sub>ij</sub>

   - If > *y*<sub>cut</sub>, stop clustering
   - Otherwise recombine *i* and *j*, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber ’91
**inclusive $k_t$ algorithm**

- Introduce angular radius $R$ (NB: dimensionless!)

$$d_{ij} = \min(p^2_{ti}, p^2_{tj}) \frac{\Delta R^2_{ij}}{R^2}, \quad d_{iB} = p^2_{ti} \quad [\Delta R^2_{ij} = (y_i - y_j)^2 + (\phi_i - \phi_j)^2]$$

- 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 $y, \phi$ cylinder.

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

\( k_t \text{ alg.: } \) Find smallest of

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

If \( d_{ij} \) recombine; if \( d_{iB}, i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Sequential recombination

**k_t alg.:** Find smallest of

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

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
**Sequential recombination**

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

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
**Sequential recombination**

**$k_t$ alg.:** Find smallest of

$$d_{ij} = \min(k^2_{ti}, k^2_{tj}) \Delta R^2_{ij}/R^2, \quad d_{iB} = k^2_{ti}$$

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

**$k_t$ alg.:** Find smallest of

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

**k_t alg.:** Find smallest of

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

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Sequential recombination

**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, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Sequential recombination

**k_t alg.:** Find smallest of

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

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\[ \phi \] assumed 0 for all towers
Sequential recombination

\( k_t \text{ alg.}: \) Find smallest of

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

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers

\[p_t/\text{GeV}
\]

\[
d_{\text{min}} \text{ is } d_{ij} = 4.8967
\]

\[y
\]

\[0 1 2 3 4
\]

\[0 10 20 30 40 50 60
\]
Sequential recombination

**kₜ alg.** Find smallest of

\[ d_{ij} = \min(k^2_{ti}, k^2_{tj}) \Delta R_{ij}^2 / R^2, \quad d_{iB} = k^2_{ti} \]

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

φ assumed 0 for all towers
Sequential recombination

$p_t$/GeV

$d_{\text{min}}$ is $d_{ij} = 20.0741$

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

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

\( k_t \) alg.: Find smallest of

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

If \( d_{ij} \) recombine; if \( d_{iB}, i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Sequential recombination

**k_t alg.:** Find smallest of

\[ d_{ij} = \min(k_{t_i}^2, k_{t_j}^2) \Delta R_{ij}^2 / R^2, \quad d_{iB} = k_{t_i}^2 \]

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Sequential recombination

\[ k_t \text{ alg.}: \text{ Find smallest of} \]

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

If \( d_{ij} \) recombine; if \( d_{iB} \), \( i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Sequential recombination

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

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

$k_t$ alg.: Find smallest of

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

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

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

**$k_t$ alg.:** Find smallest of

$$d_{ij} = \min(k^2_{ti}, k^2_{tj}) \Delta R^2_{ij} / R^2, \quad d_{iB} = k^2_{ti}$$

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

**$k_t$ alg.:** Find smallest of

$$d_{ij} = \min(k_{t_i}^2, k_{t_j}^2) \Delta R_{ij}^2 / R^2, \quad d_{iB} = k_{t_i}^2$$

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

$k_t$ alg.: Find smallest of

$$d_{ij} = \min(k_{t_i}^2, k_{t_j}^2) \Delta R_{ij}^2 / R^2, \quad d_{iB} = k_{t_i}^2$$

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
**Sequential recombination**

$k_t$ alg.: Find smallest of

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

**$k_t$ alg.:** Find smallest of

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$  
\[\phi \text{ assumed 0 for all towers}\]
**Sequential recombination**

**$k_t$ alg.:** Find smallest of

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

**$k_t$ alg.:** Find smallest of

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

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

Example clustering with $k_t$ algorithm, $R = 0.7$

$\phi$ assumed 0 for all towers
Sequential recombination

**k_t alg.:** Find smallest of

\[ d_{ij} = \min(k_{t_i}^2, k_{t_j}^2) \Delta R_{ij}^2 / R^2, \quad d_{iB} = k_{t_i}^2 \]

If \( d_{ij} \) recombine; if \( d_{iB}, i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Sequential recombination

\( k_t \text{ alg.:} \) Find smallest of

\[ d_{ij} = \min(k_{t_i}^2, k_{t_j}^2) \Delta R_{ij}^2 / R^2, \quad d_{iB} = k_{t_i}^2 \]

If \( d_{ij} \) recombine; if \( d_{iB}, i \) is a jet

Example clustering with \( k_t \) algorithm, \( R = 0.7 \)

\( \phi \) assumed 0 for all towers
Unifying idea: momentum flow within a cone only marginally modified by QCD branching

But cones come in many variants

<table>
<thead>
<tr>
<th>Finding cones</th>
<th>Processing</th>
<th>Progressive Removal</th>
<th>Split–Merge</th>
<th>Split–Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seeded, Fixed (FC)</td>
<td>GetJet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CellJet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seeded, Iterative (IC)</td>
<td>CMS Cone</td>
<td>JetClu (CDF)†</td>
<td>ATLAS cone</td>
<td></td>
</tr>
<tr>
<td>Seeded, It. + Midpoints (IC_{mp})</td>
<td></td>
<td>CDF MidPoint D0 Run II cone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seedless (SC)</td>
<td></td>
<td>SISCone</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

†JetClu also has “ratcheting”
## Cone algorithms today

Unifying idea: momentum flow within a cone only marginally modified by QCD branching

But cones come in many variants

<table>
<thead>
<tr>
<th>Finding cones</th>
<th>Processing</th>
<th>Progressive Removal</th>
<th>Split–Merge</th>
<th>Split–Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seeded, Fixed (FC)</td>
<td>GetJet</td>
<td>CellJet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seeded, Iterative (IC)</td>
<td>CMS Cone</td>
<td></td>
<td>JetClu (CDF)†</td>
<td></td>
</tr>
<tr>
<td>Seeded, It. + Midpoints (IC_{mp})</td>
<td></td>
<td></td>
<td>ATLAS cone</td>
<td></td>
</tr>
<tr>
<td>Seedless (SC)</td>
<td></td>
<td></td>
<td>CDF MidPoint</td>
<td>PxCone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D0 Run II cone</td>
<td></td>
</tr>
</tbody>
</table>

† JetClu also has “ratcheting”
Unifying idea: momentum flow within a cone only marginally modified by QCD branching

**But cones come in many variants**

<table>
<thead>
<tr>
<th>Finding cones</th>
<th>Processing</th>
<th>Progressive Removal</th>
<th>Split–Merge</th>
<th>Split–Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seeded, Fixed (FC)</td>
<td>GetJet</td>
<td>CMS Cone</td>
<td>JetClu (CDF)†</td>
<td>PxCone</td>
</tr>
<tr>
<td></td>
<td>CellJet</td>
<td></td>
<td>ATLAS cone</td>
<td></td>
</tr>
<tr>
<td>Seeded, Iterative (IC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seeded, It. + Midpoints (IC₂ₚ)</td>
<td>JetClu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seedless (SC)</td>
<td></td>
<td>SISCone</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† JetClu also has “ratcheting”
One of the simpler cones

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes
- "Hardest particle" is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
- e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes
- "Hardest particle" is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes
- "Hardest particle" is collinear unsafe more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
  e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- **Sum the momenta** use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes
- "Hardest particle" is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes
- "Hardest particle" is collinear unsafe more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones  
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- **Draw cone around seed**
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes
- "Hardest particle" is collinear unsafe
  more right away...
One of the simpler cones

- Take hardest particle as seed for cone axis
- Draw cone around seed
- **Sum the momenta** use as new seed direction, iterate until stable

Convert contents into a "jet" and remove from event.

Notes
- "Hardest particle" is collinear unsafe
- more right away...
Iterative Cone, Prog Removal (IC-PR)

**One of the simpler cones**

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta **use as new seed direction**, iterate until stable
- Convert contents into a “jet” and remove from event

**Notes**
- “Hardest particle” is collinear unsafe
  - more right away...
Iterative Cone, Prog Removal (IC-PR)

Draw cone

One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- **Draw cone around seed**
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

**Notes**

- “Hardest particle” is collinear unsafe
  more right away...
One of the simpler cones

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
\[ \text{e.g. CMS iterative cone} \]

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- "Hardest particle" is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
\[ \text{e.g. CMS iterative cone} \]

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
- more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
One of the simpler cones
  e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes

- “Hardest particle” is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- "Hardest particle" is collinear unsafe
- more right away...
One of the simpler cones
  e.g. CMS iterative cone
  
  ▶ Take hardest particle as seed for cone axis
  ▶ Draw cone around seed
  ▶ Sum the momenta use as new seed direction, iterate until stable
  ▶ Convert contents into a “jet” and remove from event

Notes
  ▶ “Hardest particle” is collinear unsafe more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
  e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- "Hardest particle" is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
  e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
Jets

Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones  
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
- more right away...
One of the simpler cones
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones

\[ \text{e.g. CMS iterative cone} \]

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
  more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
\textit{e.g.} CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
- more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones
  e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe
- more right away...
Iterative Cone, Prog Removal (IC-PR)

One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes
- “Hardest particle” is collinear unsafe

More right away...
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe \( \Rightarrow \) perturbative calculations give \( \infty \).
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe → perturbative calculations give $\infty$.
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 

\begin{figure}
\centering
\includegraphics[width=0.8\textwidth]{cone_iteration.png}
\caption{Cone iteration diagram.}
\end{figure}
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe \( \Rightarrow \) perturbative calculations give \( \infty \).
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe \( \Rightarrow \) perturbative calculations give \( \infty \).
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe. Perturbative calculations give $\infty$. 

[Diagram showing distribution of $p_T$ versus rapidity with labels for cone iteration, cone axis, and cone.]
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe → perturbative calculations give $\infty$
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 
ICPR iteration issue

Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe. Perturbative calculations give $\infty$. 

**Diagram:**
- Cone iteration
- Cone axis
- Cone
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 

\[ p_T \text{ (GeV/c)} \]

\[ \text{rapidity} \]
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 

\[ \text{cone iteration} \]

\[ \text{cone axis} \]

\[ \text{cone} \]
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 

Jet 1

$p_T$ (GeV/c)

Rapidity

Cone iteration

Cone axis

Cone
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Rightarrow$ perturbative calculations give $\infty$. 
Consequences of collinear unsafety

Collinear Safe

\[ \alpha_s^n \times (\infty) \]

Infinities cancel

Collinear Unsafe

\[ \alpha_s^n \times (\infty) \]

Infinities do not cancel

Invalidates perturbation theory
Consequences of collinear unsafety

Collinear Safe

Infinities cancel

Collinear Unsafe

Infinities do not cancel

Invalidates perturbation theory
Real life does not have infinities, but pert. infinity leaves a real-life trace

\[ \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \infty \rightarrow \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \ln \frac{p_t}{\Lambda} \rightarrow \alpha_s^2 + \alpha_s^3 + \alpha_s^3 \]

BOTH WASTED

Among consequences of IR unsafety:

<table>
<thead>
<tr>
<th>Last meaningful order</th>
<th>JetClu, ATLAS cone [IC-SM]</th>
<th>MidPoint [IC_{mp}-SM]</th>
<th>CMS it. cone [IC-PR]</th>
<th>Known at</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclusive jets</td>
<td>LO</td>
<td>NLO</td>
<td>NLO</td>
<td>NLO (→ NNLO)</td>
</tr>
<tr>
<td>W/Z + 1 jet</td>
<td>LO</td>
<td>NLO</td>
<td>NLO</td>
<td>NLO</td>
</tr>
<tr>
<td>3 jets</td>
<td>none</td>
<td>LO</td>
<td>LO</td>
<td>NLO [nlojet++]</td>
</tr>
<tr>
<td>W/Z + 2 jets</td>
<td>none</td>
<td>LO</td>
<td>LO</td>
<td>NLO [MCFM]</td>
</tr>
<tr>
<td>(m_{\text{jet}}) in 2j + X</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>LO</td>
</tr>
</tbody>
</table>

NB: 50,000,000$/$£/CHF/€ investment in NLO

Multi-jet contexts much more sensitive: **ubiquitous at LHC**

And LHC will rely on QCD for background double-checks extraction of cross sections, extraction of parameters
Real life does not have infinities, but pert. infinity leaves a real-life trace

\[ \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \infty \rightarrow \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \ln \frac{p_t}{\Lambda} \rightarrow \alpha_s^2 + \alpha_s^3 + \alpha_s^3 \]

BOTH WASTED

Among consequences of IR unsafety:

<table>
<thead>
<tr>
<th>Last meaningful order</th>
<th>JetClu, ATLAS cone [IC-SM]</th>
<th>MidPoint [IC_{mp-SM}]</th>
<th>CMS it. cone [IC-PR]</th>
<th>Known at</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclusive jets</td>
<td>LO</td>
<td>NLO</td>
<td>NLO</td>
<td>NLO (→ NNLO)</td>
</tr>
<tr>
<td>W/Z + 1 jet</td>
<td>LO</td>
<td>NLO</td>
<td>NLO</td>
<td>NLO</td>
</tr>
<tr>
<td>3 jets</td>
<td>none</td>
<td>LO</td>
<td>LO</td>
<td>NLO [nlojet++]</td>
</tr>
<tr>
<td>W/Z + 2 jets</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>NLO [MCFM]</td>
</tr>
<tr>
<td>(m_{\text{jet}}) in 2j + X</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>LO</td>
</tr>
</tbody>
</table>

NB: 50,000,000$/$£/CHF/€ investment in NLO

Multi-jet contexts much more sensitive: **ubiquitous at LHC**

And LHC will rely on QCD for background double-checks, extraction of cross sections, extraction of parameters
Real life does not have infinities, but pert. infinity leaves a real-life trace

$$\alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \infty \rightarrow \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \ln \frac{p_t}{\Lambda} \rightarrow \alpha_s^2 + \alpha_s^3 + \alpha_s^3$$

**BOTH WASTED**

Among consequences of IR unsafety:

<table>
<thead>
<tr>
<th>Last meaningful order</th>
<th>JetClu, ATLAS cone [IC-SM]</th>
<th>MidPoint [IC_{mp-SM}]</th>
<th>CMS it. cone [IC-PR]</th>
<th>Known at</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclusive jets</td>
<td>LO</td>
<td>NLO</td>
<td>NLO</td>
<td>NLO (→ NNLO)</td>
</tr>
<tr>
<td>W/Z + 1 jet</td>
<td>LO</td>
<td>NLO</td>
<td>NLO</td>
<td>NLO</td>
</tr>
<tr>
<td>3 jets</td>
<td>none</td>
<td>LO</td>
<td>LO</td>
<td>NLO [nlojet++]</td>
</tr>
<tr>
<td>W/Z + 2 jets</td>
<td>none</td>
<td>LO</td>
<td>LO</td>
<td>NLO [MCFM]</td>
</tr>
<tr>
<td>$m_{\text{jet}}$ in $2j + X$</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>LO</td>
</tr>
</tbody>
</table>

NB: 50,000,000$/$£/CHF/€ investment in NLO

Multi-jet contexts much more sensitive: **ubiquitous at LHC**

And LHC will rely on QCD for background double-checks

extraction of cross sections, extraction of parameters
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)

(As) cone algorithms give circular jets in $y - \phi$ plane. Much appreciated by experiments e.g. for acceptance corrections.
Jets

**Cone (ICPR)**

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

Much appreciated by experiments e.g. for acceptance corrections

$k_t$ 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 $k_t$
Jet algorithm characteristics:

**Cone (ICPR)**

*It. Cone (IC-PR), R=1 [FastJet]*

(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 $k_t$.
How do we solve cone IR safety problems?

Fix stable-cone finding
   ↓
   SISCone
   GPS & Soyez '07
   Same family as Tev. Run II alg

Invent "cone-like" alg.
   ↓
   anti-kt
   Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2) \Delta R_{ij}^2 \quad \longrightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour
Privilege collinear divergence over soft divergence

Cacciari, GPS & Soyez ’08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour

\[ \text{divergence over soft divergence} \]

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{t_i}^2, k_{t_j}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{t_i}^2, k_{t_j}^2)} \]

Hard stuff clusters with nearest neighbour

Privilege collinear divergence over soft divergence

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour

\[ \text{Privilege collinear divergence over soft divergence} \]

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-} k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour

divergence over soft divergence

Cacciari, GPS & Soyez '08

\begin{figure}
\centering
\includegraphics[width=\textwidth]{anti-kappa.png}
\end{figure}
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \textbf{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour

Privilege collinear divergence over soft divergence

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{t_i}^2, k_{t_j}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{t_i}^2, k_{t_j}^2)} \]

Hard stuff clusters with nearest neighbour

\[ \text{divergence over soft divergence} \]

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

**Soft stuff clusters with nearest neighbour**

\[
k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}
\]

**Hard stuff clusters with nearest neighbour**

- Privilege collinear divergence over soft divergence

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k^2_{ti}, k^2_{tj}) \Delta R^2_{ij} \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R^2_{ij}}{\max(k^2_{ti}, k^2_{tj})} \]

Hard stuff clusters with nearest neighbour

Privilege collinear divergence over soft divergence

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k^2_{ti}, k^2_{tj}) \Delta R^2_{ij} \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R^2_{ij}}{\max(k^2_{ti}, k^2_{tj})} \]

Hard stuff clusters with nearest neighbour

Divergence over soft divergence

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour

Privilege collinear divergence over soft divergence

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2) \Delta R_{ij}^2 \quad \rightarrow \quad \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour

divergence over soft divergence

Cacciari, GPS & Soyez '08
Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

\[ k_t: \quad d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \rightarrow \text{anti-}k_t: \quad d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)} \]

Hard stuff clusters with nearest neighbour

\[ \text{divergence over soft divergence} \]

Cacciari, GPS & Soyez '08

anti-\(k_t\) gives cone-like jets without using stable cones
There is plenty more choice for (IR safe) jet finding (4 good algs are Cam/Aachen, anti-$k_t$, SISCon and $k_t$)

Do all you can to avoid IR unsafe jet algorithms (ATLAS iterative cone, CMS iterative cone, etc.).

Think about the choice of parameters in your jet definition (what radius for what problem?)
Searching for high-$p_t$ (boosted) heavy particles, such as a Higgs boson.

Because LHC will have $\sqrt{s} \gg m_H$, highly boosted Higgses, $p_{tH} \gg m_H$, are not so rare.

The boost factor collimates the Higgs decay into a single jet. Can we still identify it?
$pp \rightarrow ZH \rightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

Cluster event, C/A, R=1.2
$pp \rightarrow ZH \rightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

Fill it in, → show jets more clearly
Jets

\[ pp \rightarrow ZH \rightarrow \nu\bar{\nu}b\bar{b}, \text{ @14 TeV, } m_H = 115 \text{ GeV} \]

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

Consider hardest jet, \( m = 150 \text{ GeV} \)

SIGNAL

\[ 200 < p_t Z < 250 \text{ GeV} \]

Zbb BACKGROUND

\[ 200 < p_t Z < 250 \text{ GeV} \]

arbitrary norm.
$pp \rightarrow ZH \rightarrow \nu \bar{\nu} b \bar{b}$, @14 TeV, $m_H = 115$ GeV

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

split: $m = 150$ GeV, $\frac{\max(m_1,m_2)}{m} = 0.92 \rightarrow$ repeat
$pp \rightarrow ZH \rightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

split: $m = 139$ GeV, $\frac{\max(m_1,m_2)}{m} = 0.37 \rightarrow$ mass drop
QCD lecture 4 (p. 36)

\[ pp \rightarrow ZH \rightarrow \nu\bar{\nu}b\bar{b}, \oplus 14 \text{ TeV}, \ m_H = 115 \text{ GeV} \]

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

\[ y_{12} \approx \frac{p_{t2}}{p_{t1}} \approx 0.7 \rightarrow \text{OK} + 2 \text{ b-tags (anti-QCD)} \]
$pp \rightarrow ZH \rightarrow \nu \bar{\nu} b \bar{b}$, @14 TeV, $m_H = 115$ GeV

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

$R_{\text{filt}} = 0.3$
\[ pp \rightarrow ZH \rightarrow \nu \bar{\nu} b \bar{b}, \ 14 \text{ TeV}, \ m_H = 115 \text{ GeV} \]

Herwig 6.510 + Jimmy 4.31 + FastJet 2.3

\[ R_{\text{filt}} = 0.3: \text{ take 3 hardest, } m = 117 \text{ GeV} \]
To conclude
What kinds of searches?

New resonance (e.g. $Z'$) where you see all decay products and reconstruct an invariant mass

QCD may:
- swamp signal
- smear signal

leptonic case easy; hadronic case harder
What kinds of searches?

New resonance (e.g. R-parity conserving SUSY), where undetected new stable particle escapes detection.

Reconstruct only *part* of an invariant mass → kinematic edge.

QCD may:
- swamp signal
- smear signal
Unreconstructed SUSY cascade. Study effective mass (sum of all transverse momenta).

Broad excess at high mass scales.

Knowledge of backgrounds is crucial in declaring discovery.

QCD is one way of getting handle on background.
What kinds of searches?
If you want to find out more

Classic references
QCD and collider physics
Ellis, Stirling & Webber,
Cambridge University Press 1996

The Handbook of Perturbative QCD,
the CTEQ Collaboration
http://www.phys.psu.edu/~cteq/

Advanced topics
Monte Carlos, Matching, Heavy-quarks, Jets, PDFs, etc.
E.g.: transparencies from CTEQ-MCNet 2008 QCD school
http://tr.im/oUWG