Jet algorithms

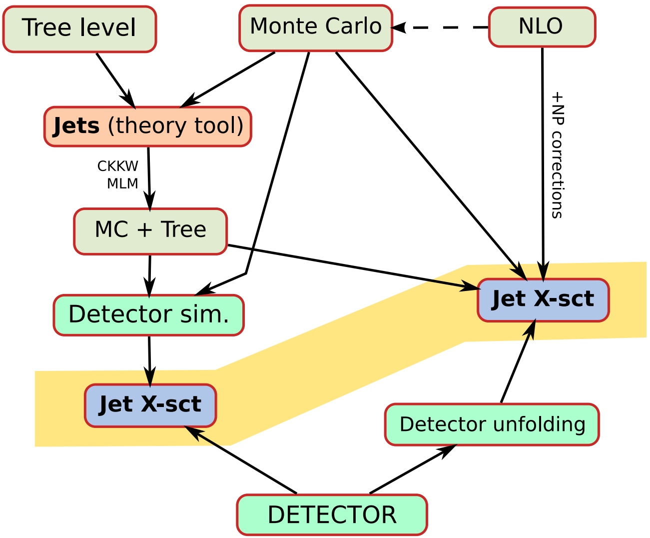
Gavin P. Salam

LPTHE, UPMC Paris 6 & CNRS

CMS JetMet meeting
CERN, Geneva
27 March 2008

Based on work with M. Cacciari (LPTHE) & G. Soyez (BNL)

QCD flowchart



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

This talk

- Jet algorithms and infrared & collinear safety
- Pileup subtraction

Cone basics I: IC-SM

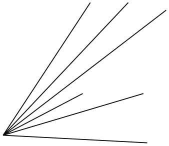
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

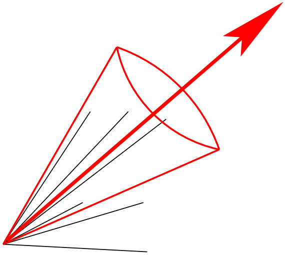
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

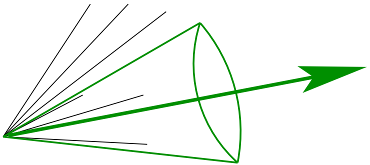
Many cone algs have two main steps:

- Find some/all stable cones

  $\equiv$ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

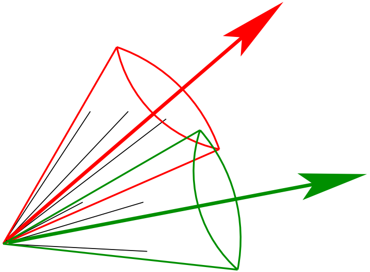
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

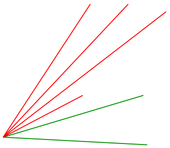
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



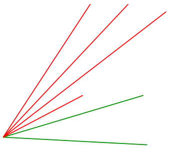
Cone basics I: IC-SM

Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents
- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Qu: How do you find the stable cones?

Until recently used iterative methods:

- use each particle as a starting direction for cone; use sum of contents as new starting direction; repeat.

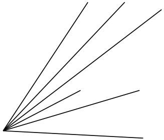
Iterative Cone with Split Merge (IC-SM)

e.g. Tevatron cones (JetClu, midpoint)
ATLAS cone

Cone basics II: IC-PR

Other cones avoid split-merge:

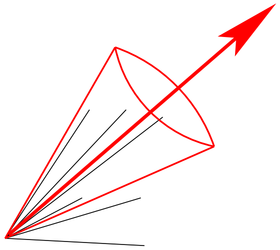
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet; remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

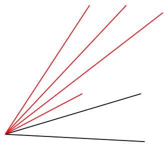
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet; remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

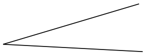
- Find one stable cone E.g. by iterating from hardest seed particle
- Call it a jet; remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

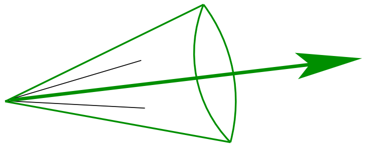
- Find one stable cone E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

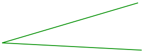
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

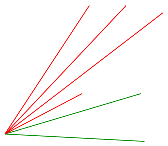
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat

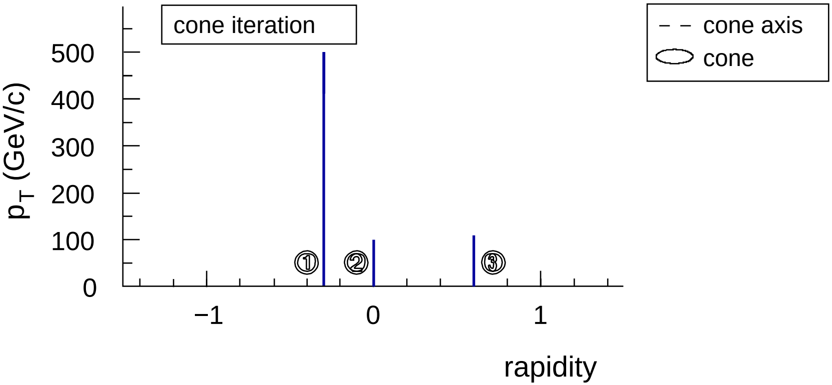


- This is not the same algorithm
- Many physics aspects differ

Iterative Cone with Progressive Removal (IC-PR)

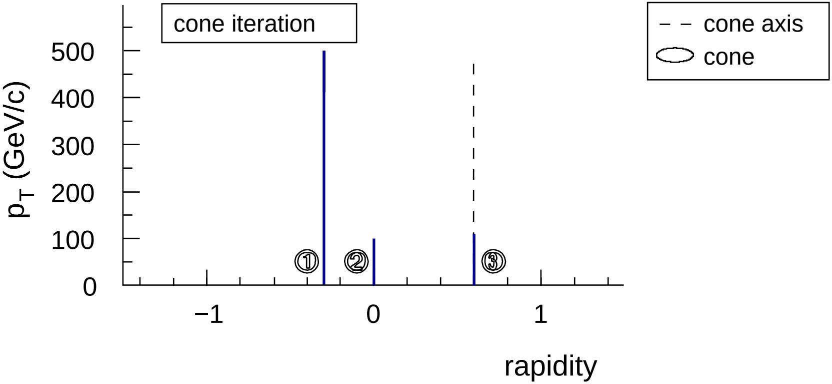
e.g. CMS it. cone, [Pythia Cone, GetJet], . . .

Iteration example



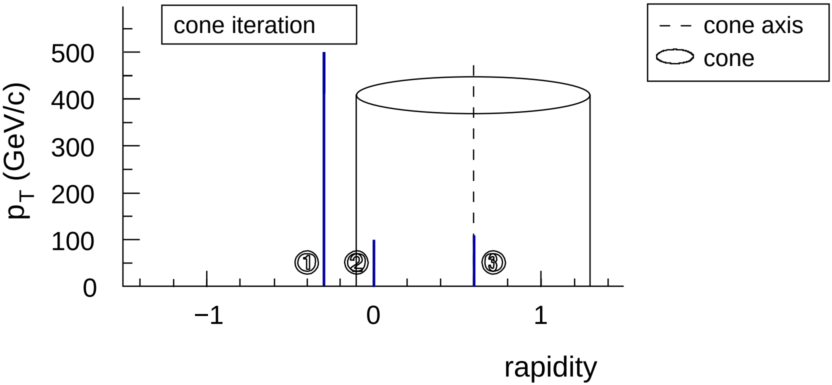
[These and related figures adapted/copied from Markus Wobisch]

Iteration example



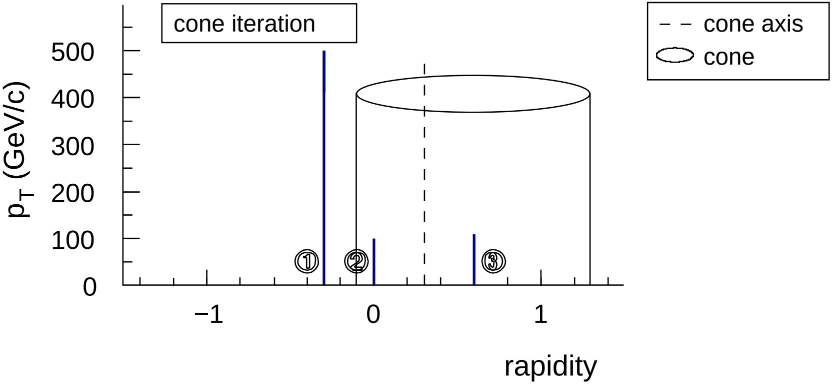
[These and related figures adapted/copied from Markus Wobisch]

Iteration example



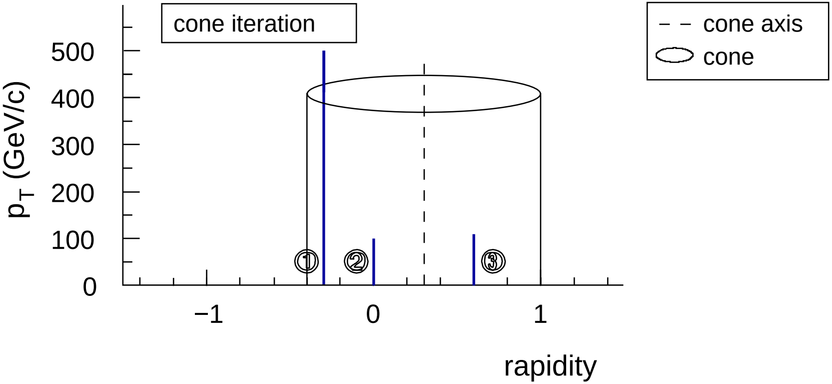
[These and related figures adapted/copied from Markus Wobisch]

Iteration example



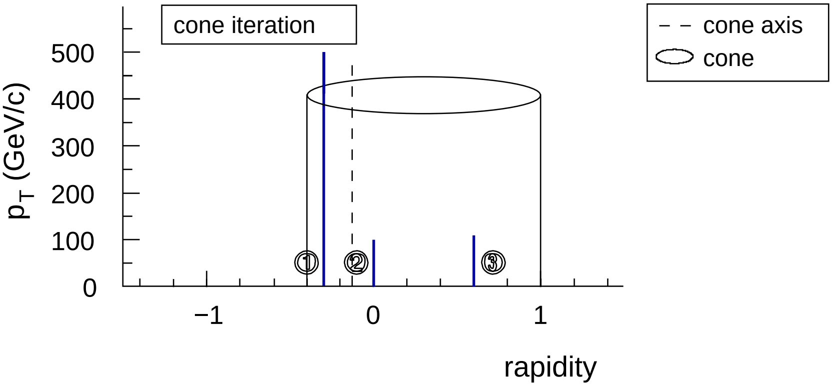
[These and related figures adapted/copied from Markus Wobisch]

Iteration example



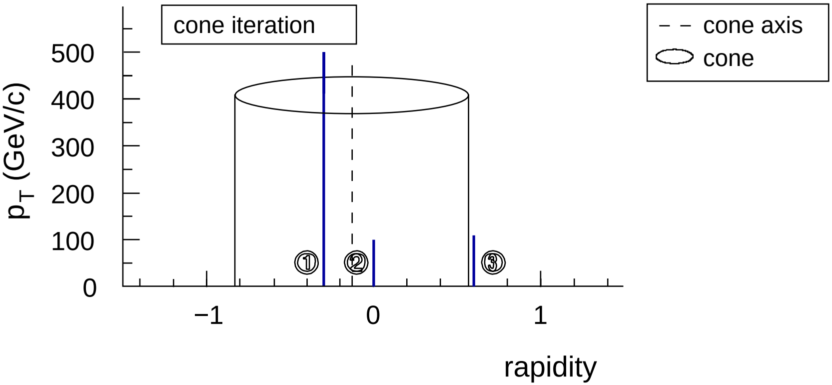
[These and related figures adapted/copied from Markus Wobisch]

Iteration example



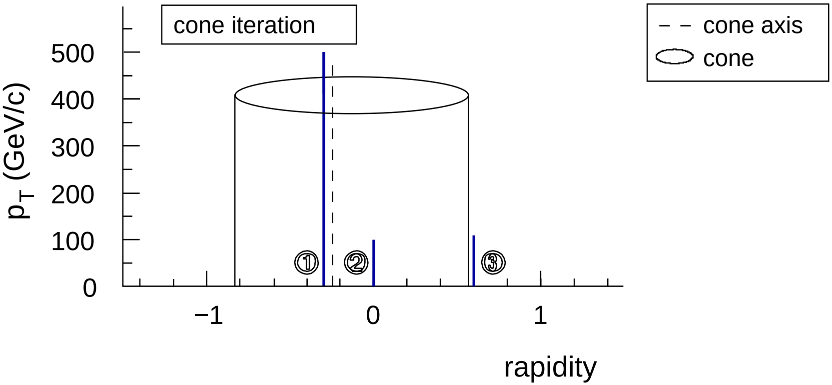
[These and related figures adapted/copied from Markus Wobisch]

Iteration example



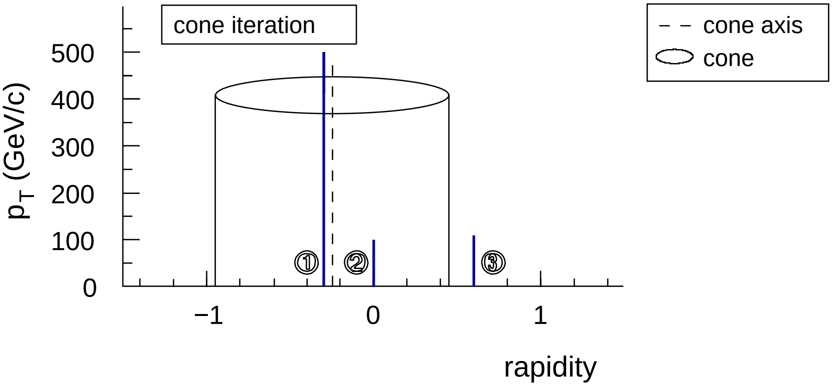
[These and related figures adapted/copied from Markus Wobisch]

Iteration example



[These and related figures adapted/copied from Markus Wobisch]

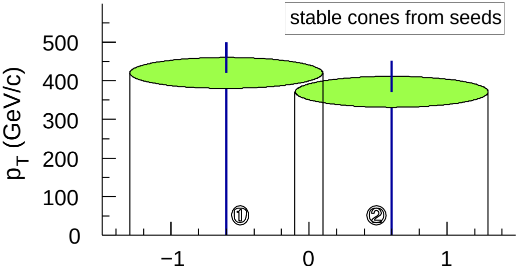
Iteration example



[These and related figures adapted/copied from Markus Wobisch]

Seeded IC-SM: infrared issue

Use of seeds is *dangerous*



Extra soft particle adds new seed → changes final jet configuration.

This is **IR unsafe**.

Kilgore & Giele '97

Partial fix: add extra seeds at midpoints of all pairs, triplets, . . . of stable cones.

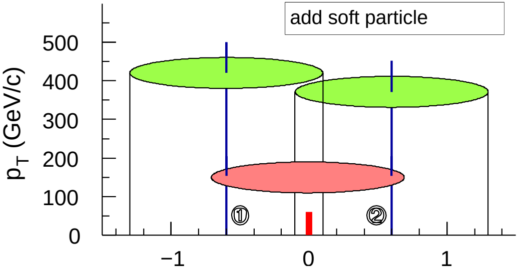
Adopted for Tevatron Run II

But only **postpones** the problem by one order . . .

Analogy: if you rely on Minuit to find minima of a function, in complex cases, results depend crucially on starting points

Seeded IC-SM: infrared issue

Use of seeds is *dangerous*



Extra soft particle adds new seed → changes final jet configuration.

This is **IR unsafe**.

Kilgore & Giele '97

Partial fix: add extra seeds at midpoints of all pairs, triplets, . . . of stable cones.

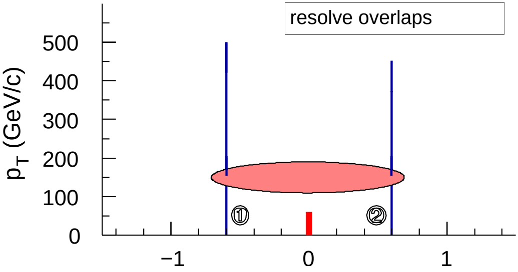
Adopted for Tevatron Run II

But only **postpones** the problem by one order . . .

Analogy: if you rely on Minuit to find minima of a function, in complex cases, results depend crucially on starting points

Seeded IC-SM: infrared issue

Use of seeds is *dangerous*



Extra soft particle adds new seed → changes final jet configuration.

This is **IR unsafe**.

Kilgore & Giele '97

Partial fix: add extra seeds at midpoints of all pairs, triplets, . . . of stable cones.

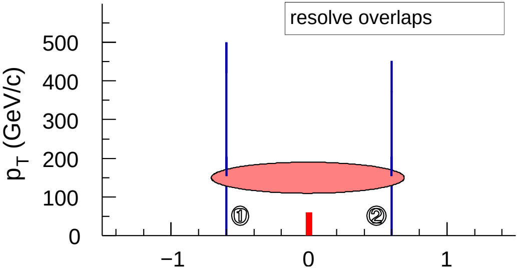
Adopted for Tevatron Run II

But only **postpones** the problem by one order . . .

Analogy: if you rely on Minuit to find minima of a function, in complex cases, results depend crucially on starting points

Seeded IC-SM: infrared issue

Use of seeds is *dangerous*



Extra soft particle adds new seed → changes final jet configuration.

This is **IR unsafe**.

Kilgore & Giele '97

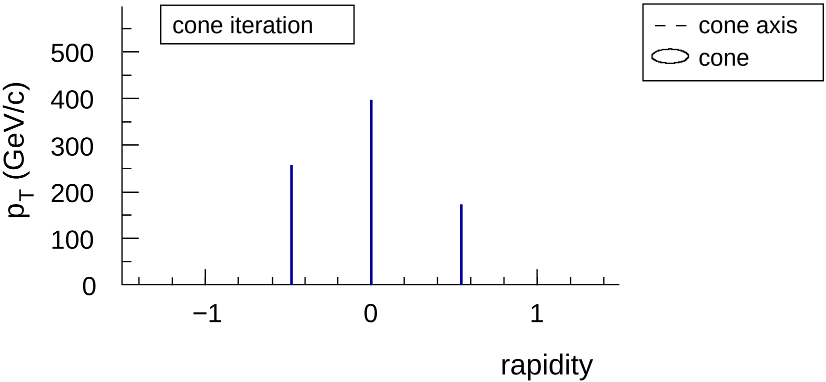
Partial fix: add extra seeds at midpoints of all pairs, triplets, . . . of stable cones.

Adopted for Tevatron Run II

But only **postpones** the problem by one order . . .

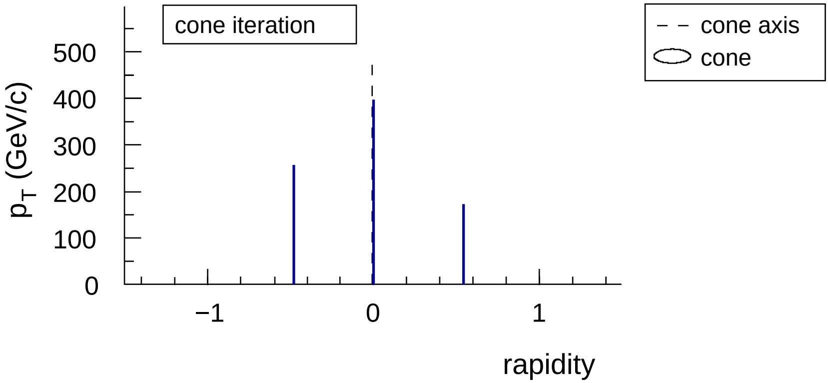
Analogy: if you rely on Minuit to find minima of a function, in complex cases, results depend crucially on starting points

ICPR iteration issue



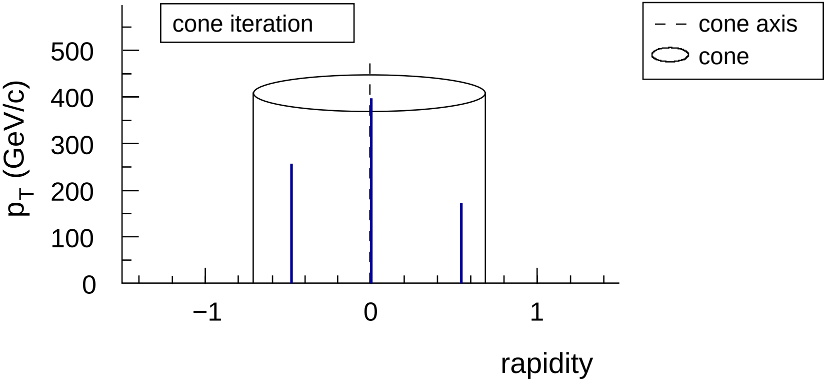
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



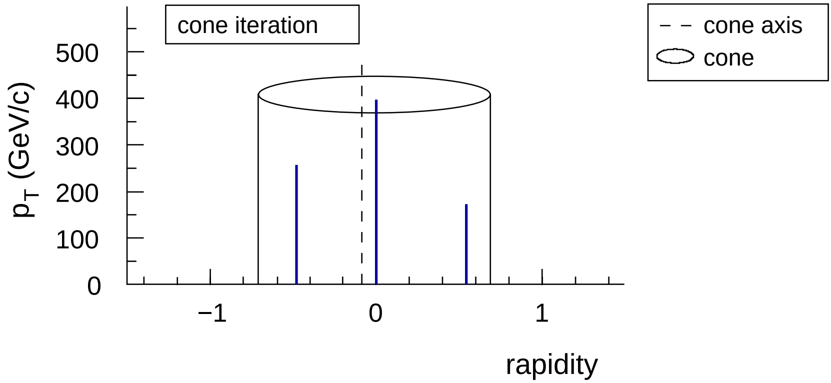
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



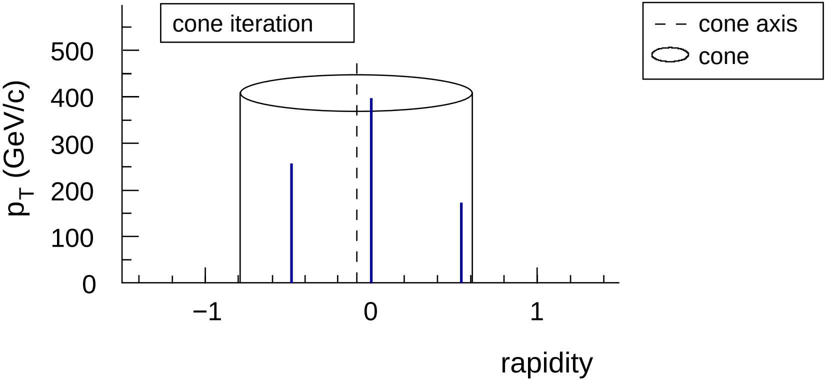
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



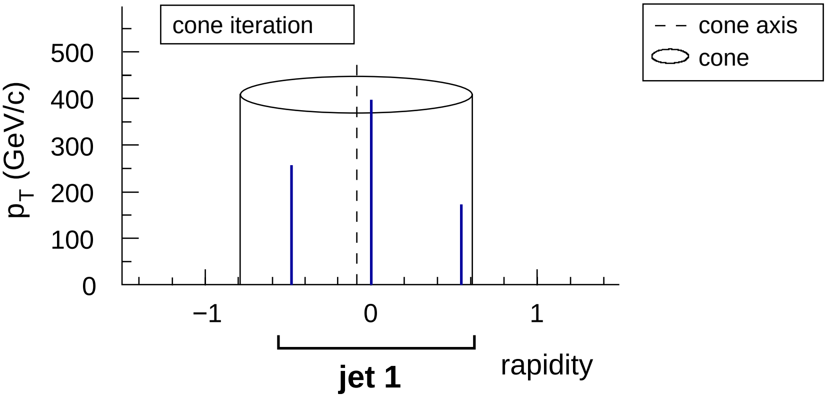
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



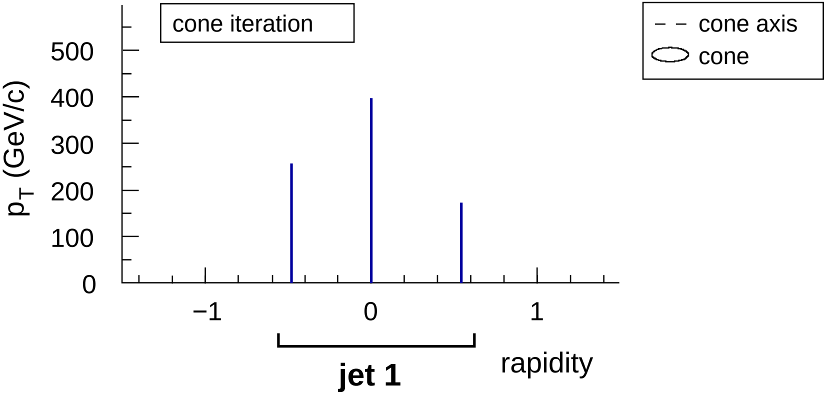
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



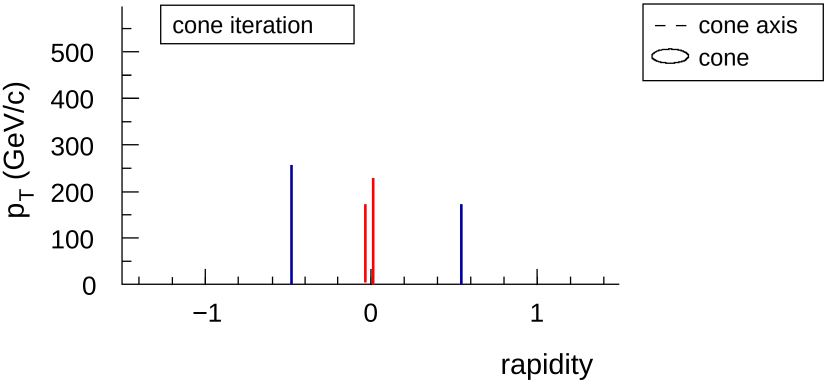
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



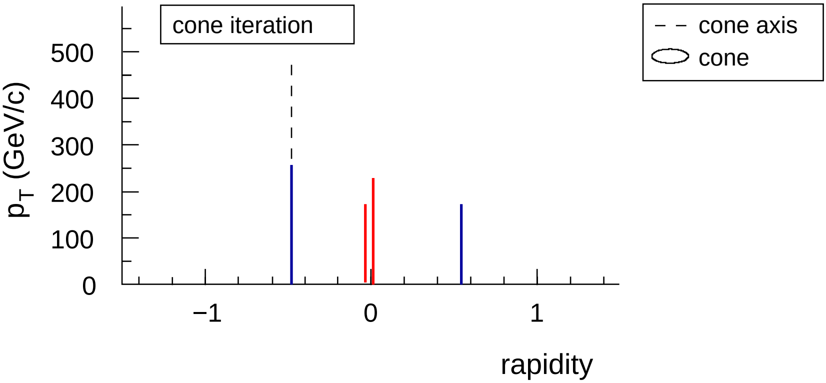
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



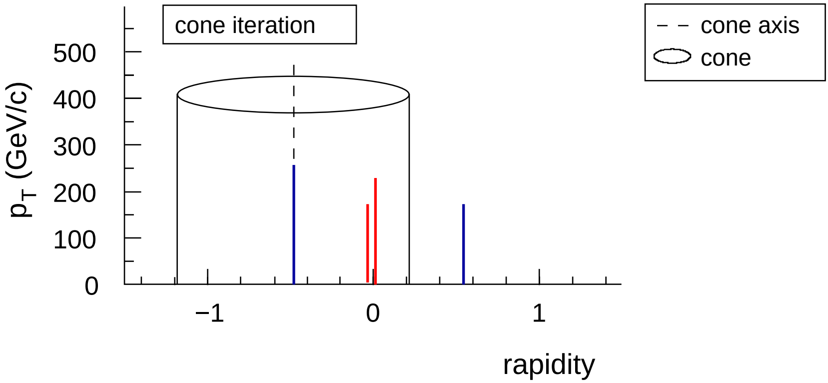
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



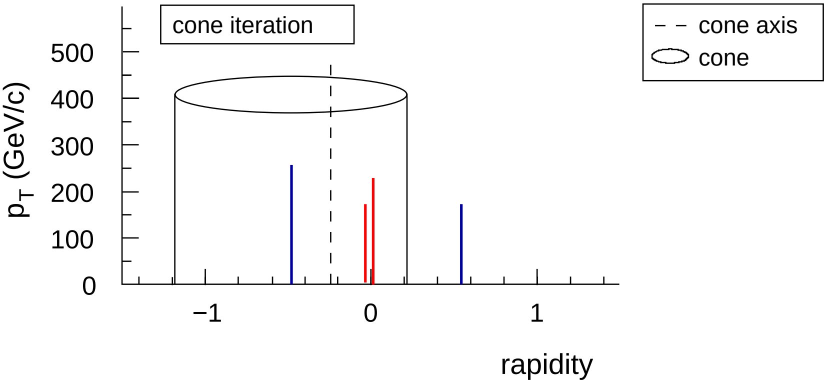
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



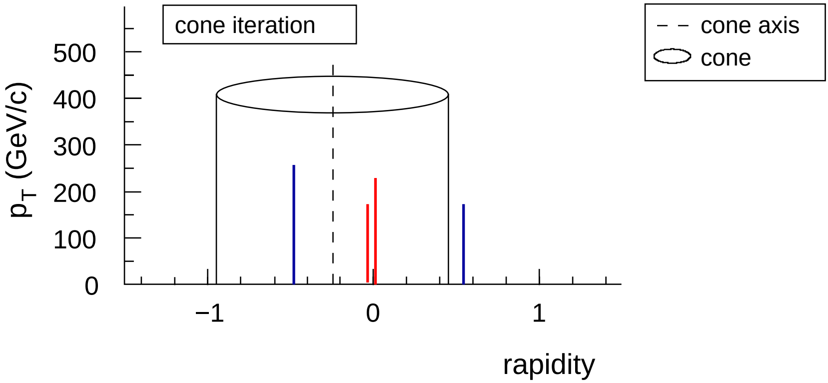
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



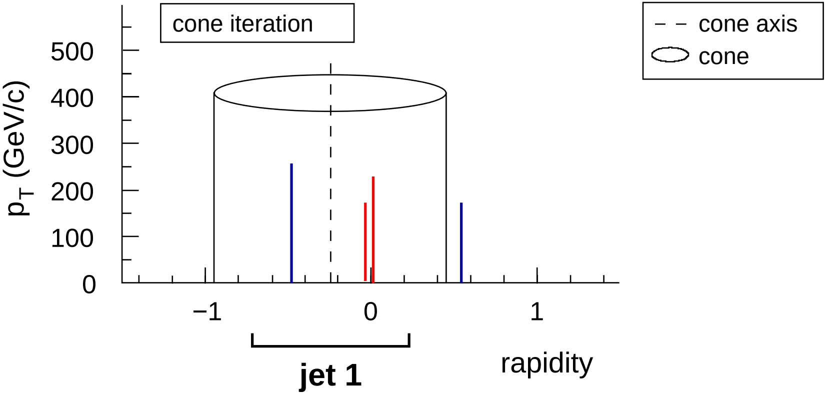
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



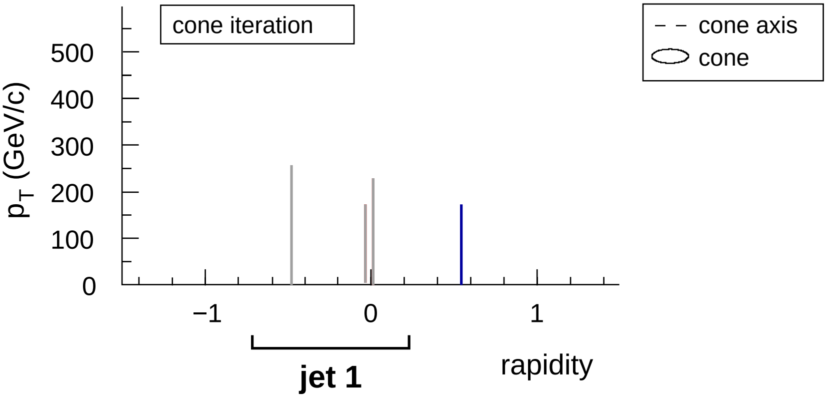
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



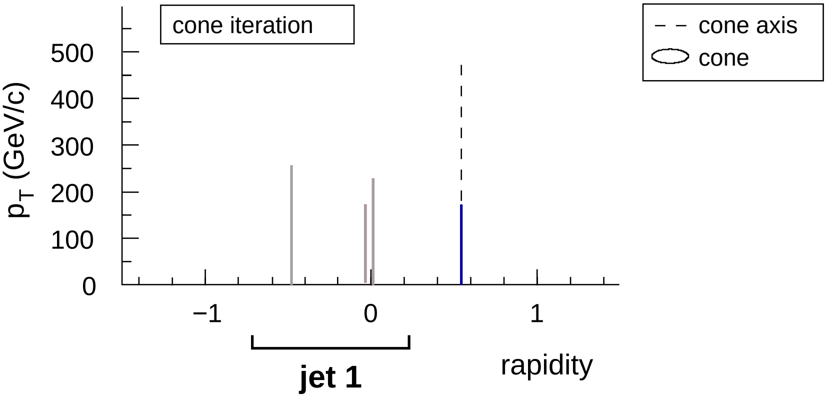
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



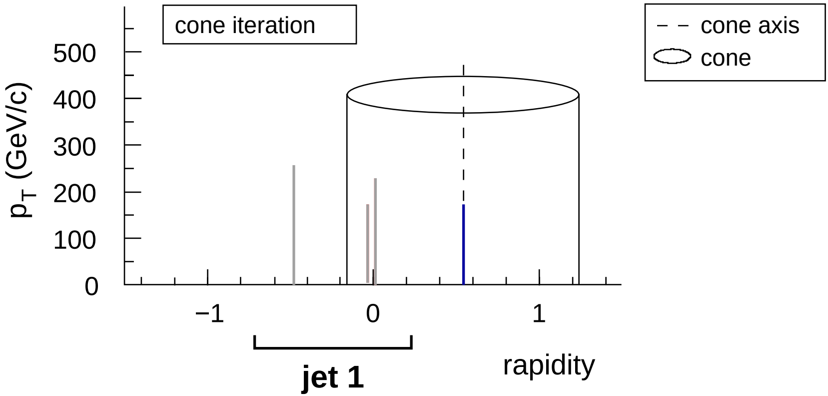
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



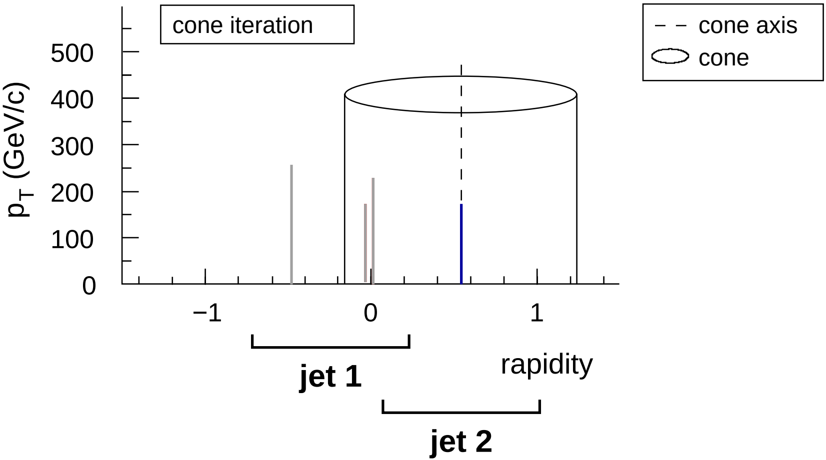
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



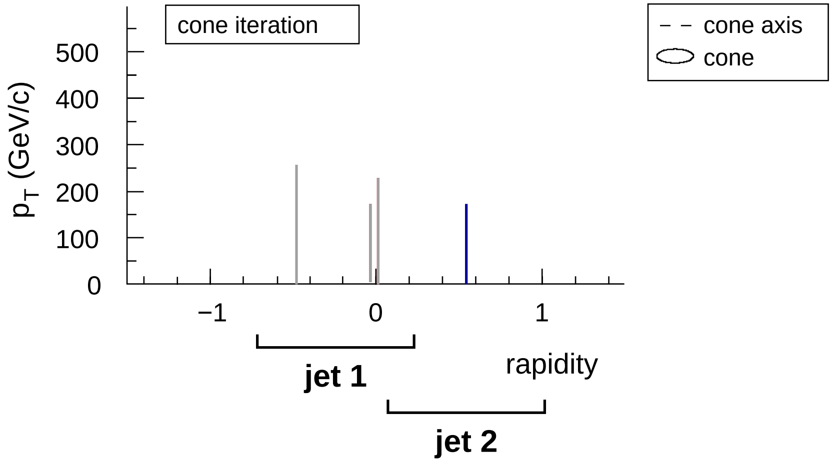
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



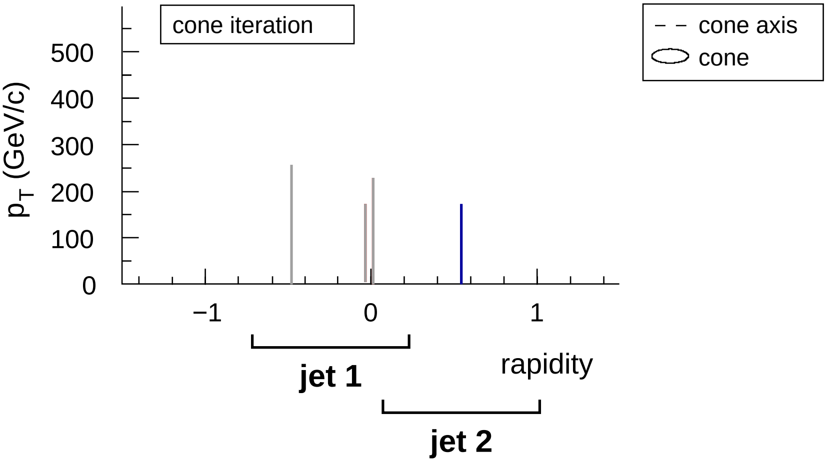
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

ICPR iteration issue



Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe

Infrared and Collinear Safety

Snowmass Accord (1990):

FERMILAB-Conf-90/249-E
[E-741/CDF]

Toward a Standardization of Jet Definitions

Several important properties that should be met by a jet definition are [3]:

1. Simple to implement in an experimental analysis;
2. Simple to implement in the theoretical calculation;
3. Defined at any order of perturbation theory;
4. Yields finite cross section at any order of perturbation theory;
5. Yields a cross section that is relatively insensitive to hadronization.

Property 4 ≡ Infrared and Collinear (IRC) Safety. It helps ensure:

- Non-perturbative effects are suppressed by powers of Λ_{QCD}/p_t
- Each order of perturbation theory is smaller than previous (at high p_t)

Infrared and Collinear Safety

Snowmass Accord (1990):

FERMILAB-Conf-90/249-E
[E-741/CDF]

Toward a Standardization of Jet Definitions *

Several important properties that should be met by a jet definition are [3]:

1. Simple to implement in an experimental analysis;
2. Simple to implement in the theoretical calculation;
3. Defined at any order of perturbation theory;
4. Yields finite cross section at any order of perturbation theory;
5. Yields a cross section that is relatively insensitive to hadronization.

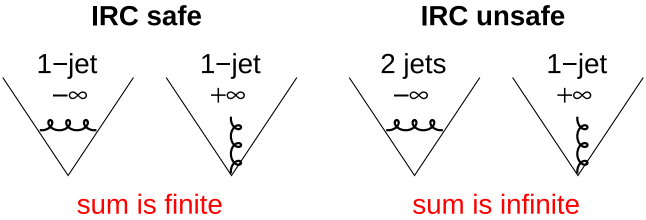
Property 4 $\equiv$ Infrared and Collinear (IRC) Safety. It helps ensure:

- Non-perturbative effects are suppressed by powers of Λ_{QCD}/p_t
- Each order of perturbation theory is smaller than previous (at high p_t)

IRC safety crucial for theory

Soft emission, collinear splitting are both **infinite** in pert. QCD.

Infinities **cancel** with loop diagrams if jet-alg IRC safe



Some calculations simply become **meaningless**

IRC saftey & real-life

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

Among consequences of IR unsafety:

	Last meaningful order			
	ATLAS cone [IC-SM]	MidPoint [IC$_{mp}$-SM]	CMS it. cone [IC-PR]	Known at
Inclusive jets	LO	NLO	NLO	NLO (→ NNLO)
$W/Z + 1$ jet	LO	NLO	NLO	NLO
3 jets	**none**	LO	LO	NLO [nlojet++]
$W/Z + 2$ jets	**none**	LO	LO	NLO [MCFM]
m_{jet} in $2j + X$	**none**	**none**	**none**	LO

NB: \$30 – 50M investment in NLO

Note: simple environments (e.g. dijets) suffer less ("a jet is a jet").

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

And you'll rely on QCD for background double-checks
extraction of cross sections, extraction of parameters

IRC safety not just for theory

1. Detectors play tricks with soft particles

   calorimeter thresholds
   magnetic fields acting on charged particles
   calorimeter noise

2. Detectors split/merge collinear particles

   Two particles into single calo-tower
   One particles showers into two calo-towers

3. High lumi adds lots of extra soft seeds

IRC safety provides resilience to these effects
1 & 3 shift energy scale, but don't change overall jet-structure

If jet-algorithm is not IRC safe, fine-details of detector effects have potentially significant impact

IRC safety not just for theory

1. Detectors play tricks with soft particles

   calorimeter thresholds

   magnetic fields acting on charged particles

   calorimeter noise

2. Detectors split/merge collinear particles

   Two particles into single calo-tower

   One particles showers into two calo-towers

3. High lumi adds lots of extra soft seeds

IRC safety provides resilience to these effects

1 & 3 shift energy scale, but don't change overall jet-structure

If jet-algorithm is not IRC safe, fine-details of detector effects have potentially significant impact

Our logic re Snowmass/IRC safety

- IRC safety is non-negotiable
  - It's part of why jets were defined originally — Sterman-Weinberg '77
  - It's essential for theory calculations to make sense
  - This is a consensus view — or at least, has been affirmed by every major "jet-workshop" since 1991. — Snowmass '91, Run II '00, Tev4LHC '06, Les Houches '07

- **But:** some IRC unsafe algorithms might have other "nice" properties
  - especially low UE sensitivity
  - circularity of jets

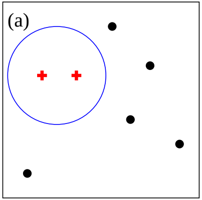
So let's keep those nice properties, but engineer away the IRC unsafety. — May require non-obvious approaches

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

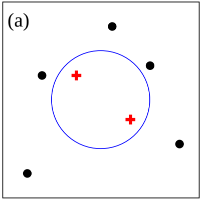
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

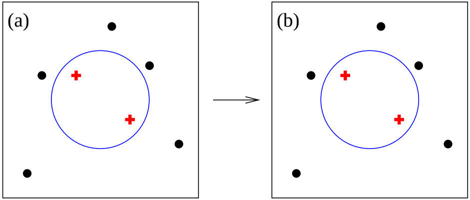
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

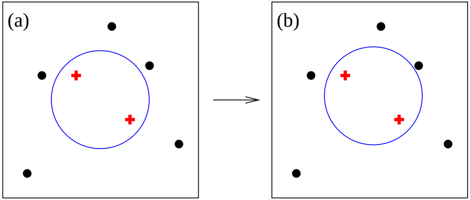
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

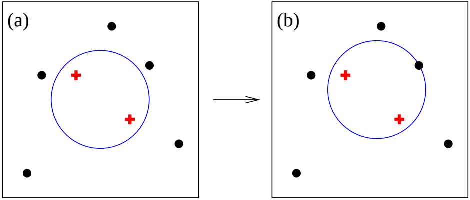
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

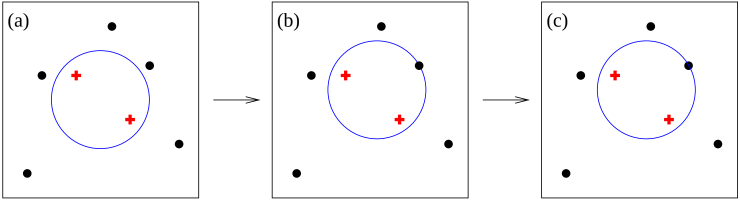
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

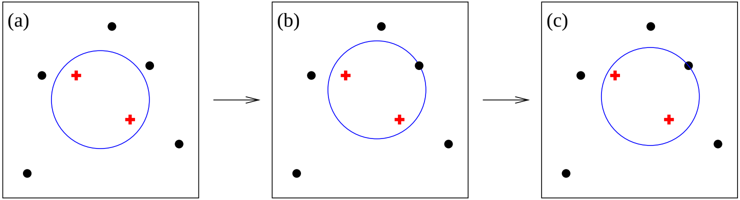
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

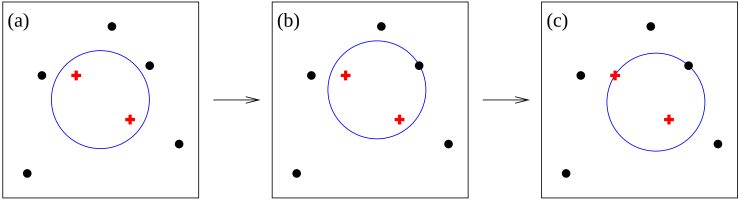
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

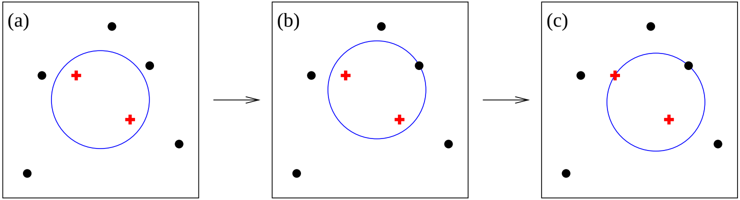
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

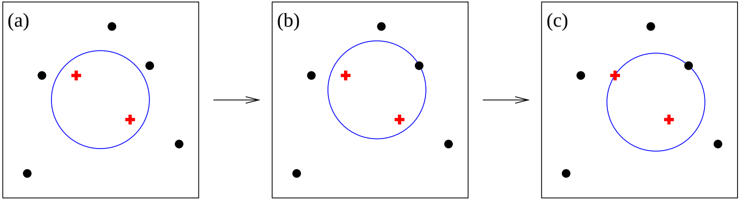
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

GPS & Soyez '07

SISCone algorithm as a whole

1: Put the set of current particles equal to the set of all particles in the event.

2: **repeat**

3: Find *all* stable cones of radius $\boldsymbol{R}$ for the current set of particles, e.g. using algorithm 2.

4: For each stable cone, create a protojet from the current particles contained in the cone, and add it to the list of protojets.

5: Remove all particles that are in stable cones from the list of current particles.

6: **until** No new stable cones are found, or one has gone around the loop N_{pass} times.

7: Run a Tevatron Run-II type split–merge procedure, algorithm 3, on the full list of protojets, with overlap parameter $\boldsymbol{f}$ and transverse momentum threshold $p_{t,\mathrm{min}}$.

Is it truly IR safe?

- Generate event with $2 < N < 10$ hard particles, find jets
- Add $1 < N_{soft} < 5$ soft particles, find jets again [repeatedly]
- If the jets are different, algorithm is IR unsafe.

Unsafety level	failure rate
2 hard + 1 soft	$\sim 50\%$
3 hard + 1 soft	$\sim 15\%$
SISCone	IR safe !

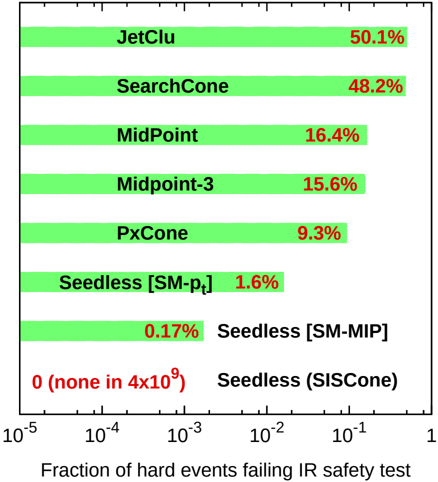
Be careful with split–merge too

Is it truly IR safe?

- Generate event with $2 < N < 10$ hard particles, find jets
- Add $1 < N_{soft} < 5$ soft particles, find jets again [repeatedly]
- If the jets are different, algorithm is IR unsafe.

Unsafety level	failure rate
2 hard + 1 soft	$\sim$ 50%
3 hard + 1 soft	$\sim$ 15%
SISCone	IR safe !

Be careful with split–merge too



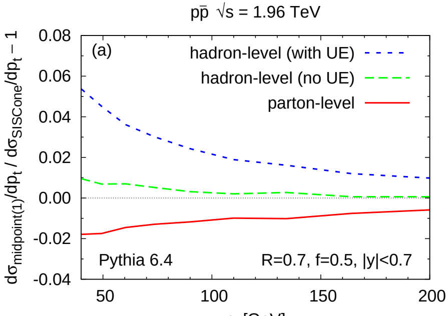
How much does IR safety *really* matter?

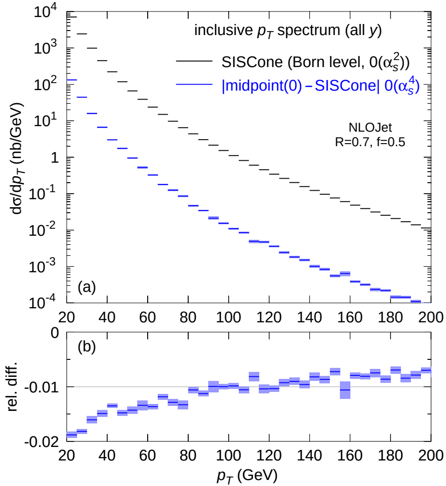
Compare midpoint and SISCone

Result depends on observable:

- inclusive jet spectrum is the least sensitive (affected at NNLO)
- larger differences (5 – 10%) at hadron level

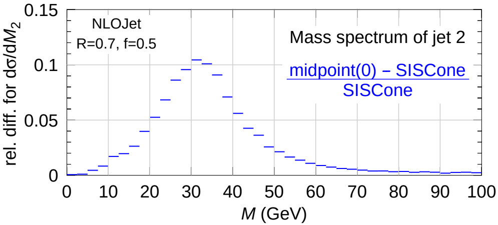
seedless reduces UE effect





IR safety & multi-jet observables

Look at jet masses in multijet events. NB: Jet masses reconstruct boosted $W/Z/H/top$ in BSM searches



Select 3-jet events

$$p_{t1,2,3} > \{120, 60, 20\}\ \text{GeV},$$

Calculate LO jet-mass spectrum for jet 2, compare midpoint with SISCone.

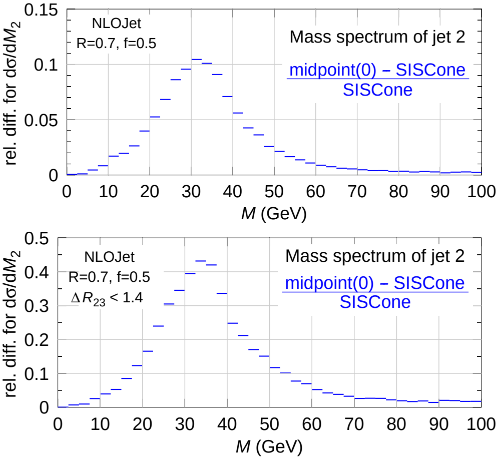
- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$

e.g. for jets from common decay chain

In complex events, IR safety matters

IR safety & multi-jet observables

Look at jet masses in multijet events. NB: Jet masses reconstruct boosted $W/Z/H/top$ in BSM searches



Select 3-jet events

$$p_{t1,2,3} > \{120, 60, 20\} \text{ GeV},$$

Calculate LO jet-mass spectrum for jet 2, compare midpoint with SISCone.

- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$

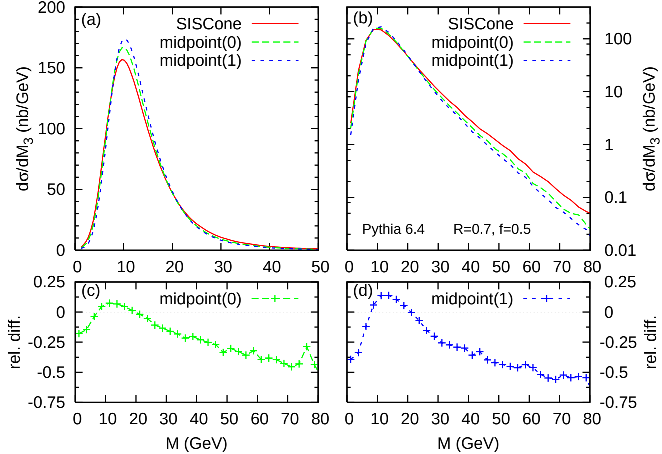
  e.g. for jets from common decay chain

In complex events, IR safety matters

Multi-jet observables: after showering

Showering puts in many extra seeds: missing stable cones (in midpoint) should be less important?

Look at 3rd jet mass distribution (no ΔR_{23} cut):



Missing stable cones → 50% effects even after showering

anti-k_t

SISCone is good replacement for JetClu, Atlas iterative cone, and MidPoint type cones

But these (xC-SM) all rather different from CMS It. Cone (IC-PR)

Differ @ NLO for incl. jets

Do not have area $= \pi R^2$

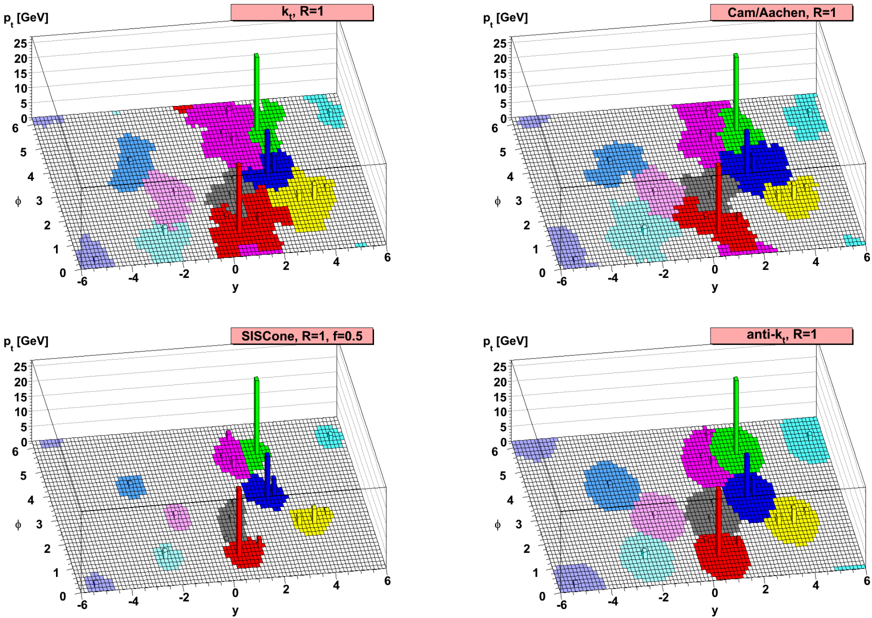
Alternative: drop the "cone" in definition, but get an algorithm that still acts like a cone: **anti-**k_t

1. Find smallest of d_{ij}, d_{iB}: $d_{ij} = \min(p_{ti}^{-2}, p_{tj}^{-2}) \Delta R_{ij}^2 / R^2 \,, \qquad d_{iB} = p_{ti}^{-2}$
2. if ij, recombine them; if iB, call i a jet, and remove from list of particles
3. repeat from step 1 until no particles left.

Cacciari, GPS & Soyez '08

Looks like k_t but behaves IC-PR.

Jet contours – visualised



Pileup subtraction

Tevatron approach:

- Measure min-bias
- subtract $n_{\rm vertex} - 1 \times \pi R^2 \times$min-bias density

Assumes jet area $= \pi R^2$; min-bias doesn't fluctuate

Used as "argument" against new jet algs

Two approaches that I see as worth thinking about:

- Subtract pileup from calorimeter, before passing information to the jet algorithm

  Issues: is calorimeter right scale to be subtracting on?

  Some towers end up being negative — how does one address this?

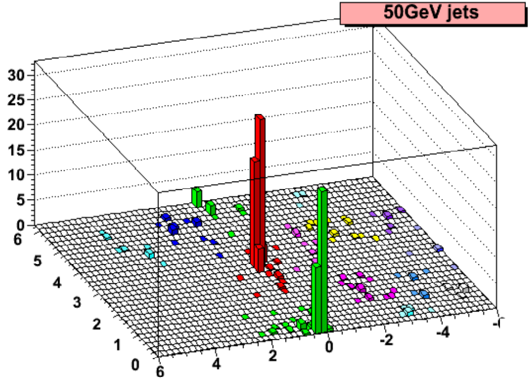
- Subtract pileup from jets, after having carried out jet finding on calorimeter that includes the pileup.

  Negative jets easily dealt with (throw them away)

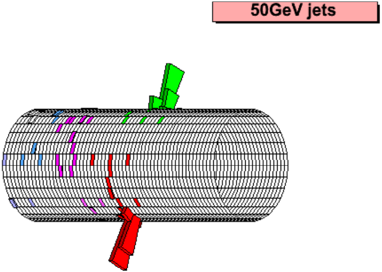
  But pileup can modify clustering (back-reaction)

Last one developed with Cacciari ('07), to show that one can subtract pileup effectively with any alg.

Jets, pileup and areas



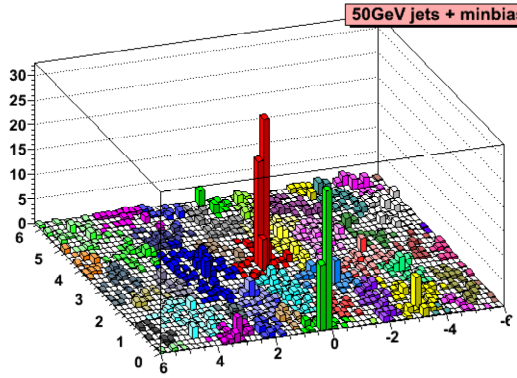
'Standard hard' event
Two well isolated jets



$\sim$ 200 particles

Clustering takes $\lesssim 1ms$

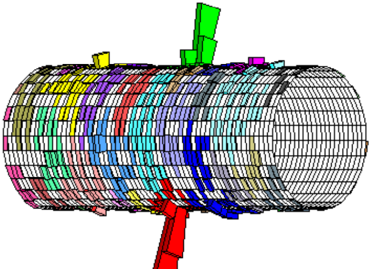
Jets, pileup and areas



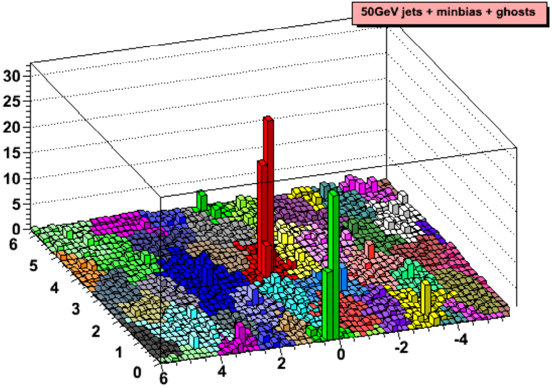
Add 10 min-bias events (moderately high lumi)

$\sim$ 2000 particles

Clustering takes $\sim 10ms$

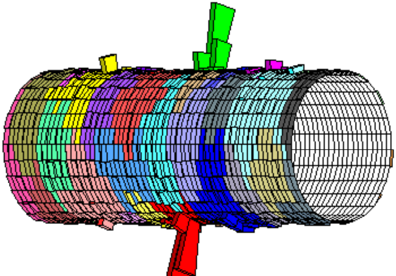


Jets, pileup and areas



Add dense coverage of infinitely soft *“ghosts”*

See how many end up in jet to measure jet area

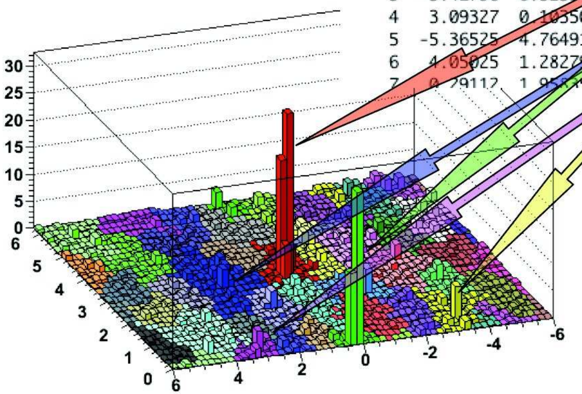


∼ 10000 particles

Clustering takes ∼ 0.2*s*

Jet areas

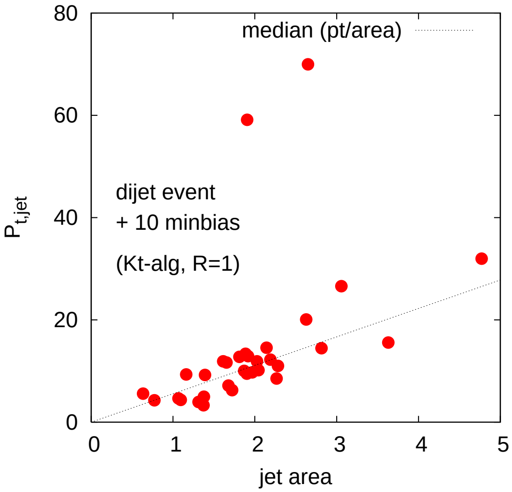
```
iev 0 (irepeat 24): number of particles = 1428
strategy used =  NlnN
number of particles = 9051
Total area: 76.0265
Expected area: 76.0265
 ijet    eta       phi          Pt          area +-   err
    0    0.15050   3.24498     69.970       2.625 +-  0.020
    1    0.18579   0.13150     59.133       1.896 +-  0.020
    2    2.33840   3.23960     31.976       4.749 +-  0.028
    3   -3.41796   0.52394     26.595       3.084 +-  0.021
    4    3.09327   0.10350     20.072       2.688 +-  0.023
    5   -5.36525   4.76491     19.593       2.780 +-  0.012
    6    4.05025   1.28279     15.561       3.592 +-  0.028
    7    0.29112   1.95835     14.566       2.114 +-  0.018
```



Approximate linear relation between Pt and area for minimum bias jets.

Can be used on an event-by-event basis to correct the hard jets

Jet areas



Jet areas in k_t algorithm are quite varied

Because k_t-alg adapts to the jet structure

- Hard jets' contamination from min-bias $\sim$ area

  Area varies even for SISCone

- Soft jets' p_t/area tells you about min-bias normalisation and fluctuations

Median p_t/area across the set of jets in an event is a good estimator of pileup+UE in *that event*

Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

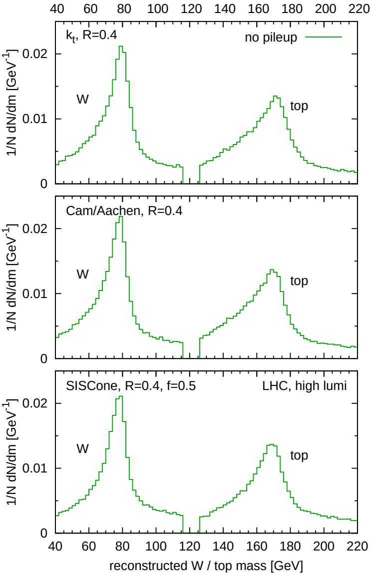
$$p_t \to p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

Same simple procedure works for a range of algorithms



Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

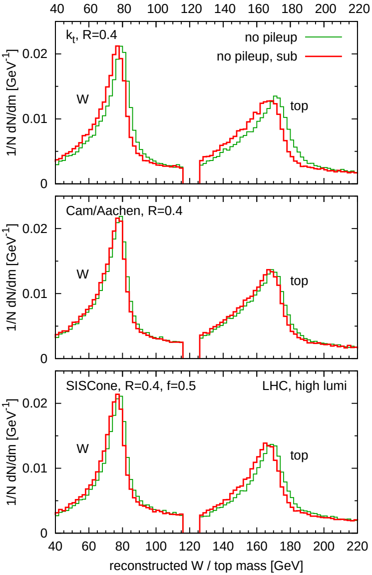
$$p_t \rightarrow p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

Same simple procedure works for a range of algorithms



Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

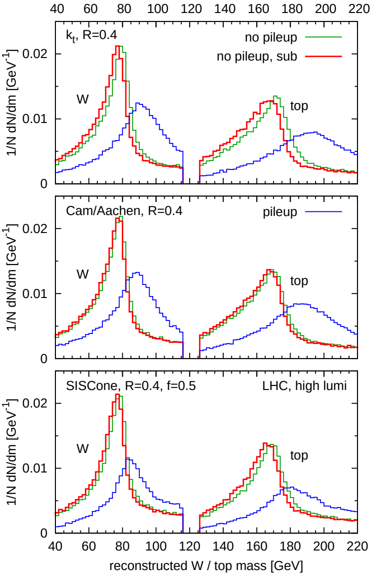
$$p_t \rightarrow p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

Same simple procedure works for a range of algorithms



Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

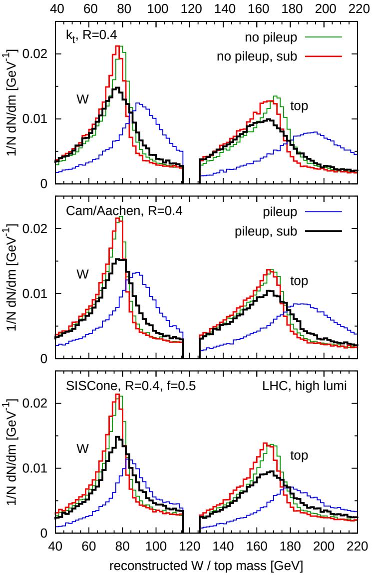
$$p_t \rightarrow p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

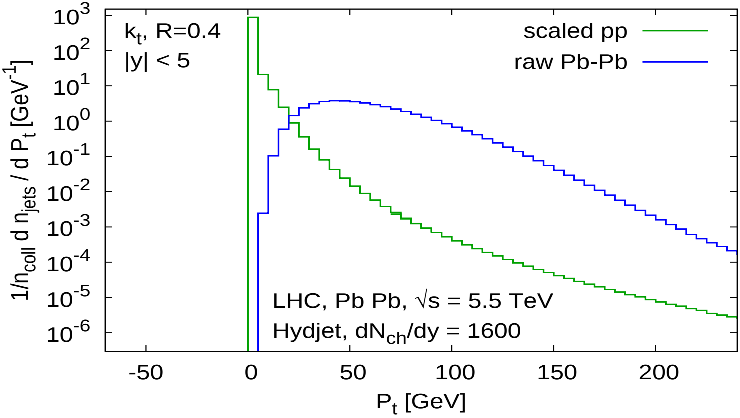
Same simple procedure works for a range of algorithms



Subtraction for Pb Pb at LHC

Example: inclusive jet spectrum

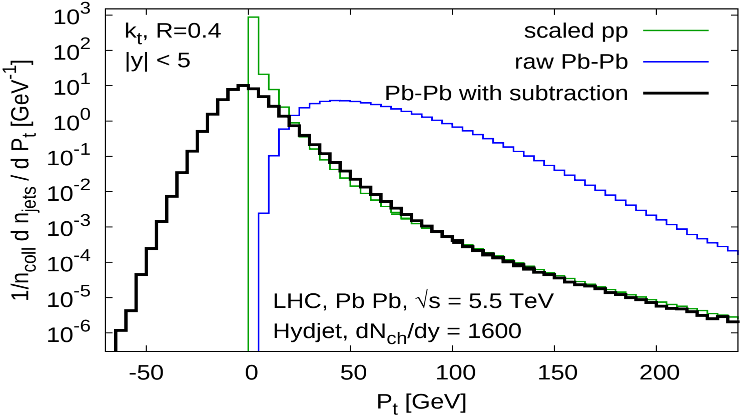
- Speed makes it easy to run k_t and Cam/Aachen on all 30k particles in HI event
- Subtraction provides a way to get sensible results, without biases from cut on low-p_t particles.



Subtraction for Pb Pb at LHC

Example: inclusive jet spectrum

- Speed makes it easy to run k_t and Cam/Aachen on all 30k particles in HI event
- Subtraction provides a way to get sensible results, without biases from cut on low-p_t particles.



Conclusions

- Different algorithms are complementary — Subject for a whole talk!
- You want to have acccess to that variety
- But you **do not** want to have IR unsafe algorithms

  Less stable, compromise benefit from $50M theory

- For each type of IRC unsafe cone alg., ∃ a sensible replacement

  IC-SM → SISCone, IC-PR → anti-k_t

- No major cost in speed — All accessible through fastjet

- You want to be able to subtract pileup *independently* from the jet algorithm
- Area-based subtraction with in-situ pileup measurement seems effective.

Conclusions

- Different algorithms are complementary — Subject for a whole talk!
- You want to have acccess to that variety
- But you **do not** want to have IR unsafe algorithms

  Less stable, compromise benefit from $50M theory

- For each type of IRC unsafe cone alg., ∃ a sensible replacement

  IC-SM → SISCone, IC-PR → anti-k_t

- No major cost in speed — All accessible through fastjet

- You want to be able to subtract pileup *independently* from the jet algorithm
- Area-based subtraction with in-situ pileup measurement seems effective.

Conclusions

- Different algorithms are complementary — Subject for a whole talk!
- You want to have acccess to that variety
- But you **do not** want to have IR unsafe algorithms

  Less stable, compromise benefit from $50M theory

- For each type of IRC unsafe cone alg., ∃ a sensible replacement

  IC-SM → SISCone, IC-PR → anti-k_t

- No major cost in speed — All accessible through fastjet

- You want to be able to subtract pileup *independently* from the jet algorithm
- Area-based subtraction with in-situ pileup measurement seems effective.

EXTRA MATERIAL

SISCone part 2: finding stable cones

For any group of collinear particles, merge them into a single particle.

for particle $i = 1 \ldots N$ **do**

  Find all particles j within a distance $2R$ of i. If there are no such particles, i forms a stable cone of its own.

  Otherwise for each j identify the two circles for which i and j lie on the circumference. For each circle, compute the angle of its centre C relative to i, $\zeta = \arctan \frac{\Delta\phi_{iC}}{\Delta y_{iC}}$.

  Sort the circles into increasing angle ζ.

  Take the first circle in this order, and call it the current circle. Calculate the total momentum and checkxor for the cones that it defines. Consider all 4 permutations of edge points being included or excluded. Call these the "current cones".

  repeat

    for each of the 4 current cones **do**

      If this cone has not yet been found, add it to the list of distinct cones.

      If this cone has not yet been labelled as unstable, establish if the in/out status of the edge particles (with respect to the cone momentum axis) is the same as when defining the cone; if it is not, label the cone as unstable.

    end for

    Move to the next circle in order. It differs from the previous one either by a particle entering the circle, or one leaving the circle. Calculate the momentum for the new circle and corresponding new current cones by adding (or removing) the momentum of the particle that has entered (left); the checkxor can be updated by XORing with the label of that particle.

  until all circles considered.

end for

for each of the cones not labelled as unstable **do**

  Explicitly check its stability, and if it is stable, add it to the list of stable cones (protojets).

end for

SISCone part 3: split–merge

1: **repeat**

Remove all protojets with $p_t < p_{t,\min}$.

Identify the protojet (i) with the highest $\tilde{p}_t$ ($\tilde{p}_{t,\mathrm{jet}} = \sum_{i\in\mathrm{jet}} |p_{t,i}|$).

Among the remaining protojets identify the one (j) with highest $\tilde{p}_t$ that shares particles (overlaps) with i.

5: **if** there is such an overlapping jet **then**

6: Determine the total $\tilde{p}_{t,\mathrm{shared}} = \sum_{k\in i\&j} |p_{t,k}|$ of the particles shared between i and j.

7: **if** $\tilde{p}_{t,\mathrm{shared}} < f\,\tilde{p}_{t,j}$ **then**

Each particle that is shared between the two protojets is assigned to the one to whose axis it is closest. The protojet momenta are then recalculated.

9: **else**

Merge the two protojets into a single new protojet (added to the list of protojets, while the two original ones are removed).

11: **end if**

12: If steps 7–11 produced a protojet that coincides with an existing one, maintain the new protojet as distinct from the existing copy(ies).

13: **else**

Add i to the list of final jets, and remove it from the list of protojets.

15: **end if**

16: **until** no protojets are left.

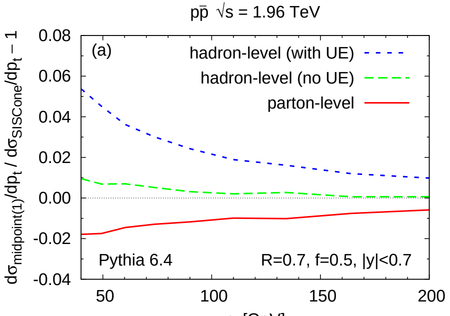
How much does IR safety *really* matter?

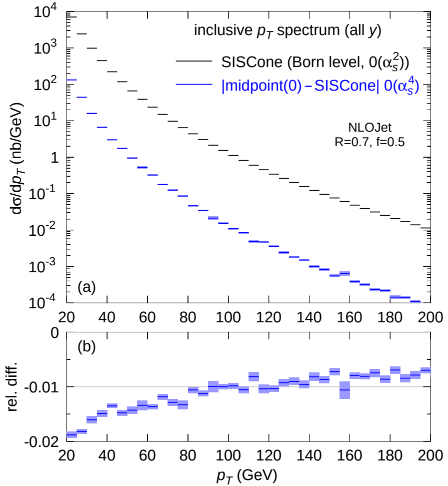
Compare midpoint and SISCone

Result depends on observable:

- inclusive jet spectrum is the least sensitive (affected at NNLO)
- larger differences (5 – 10%) at hadron level

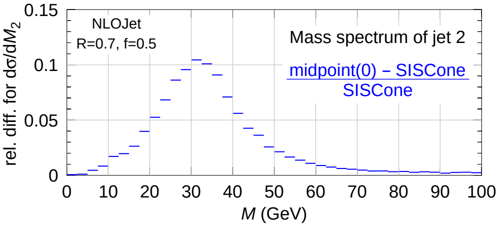
seedless reduces UE effect





IR safety & multi-jet observables

Look at jet masses in multijet events. NB: Jet masses reconstruct boosted $W/Z/H/top$ in BSM searches



Select 3-jet events

$$p_{t1,2,3} > \{120, 60, 20\}\ \text{GeV},$$

Calculate LO jet-mass spectrum for jet 2, compare midpoint with SISCone.

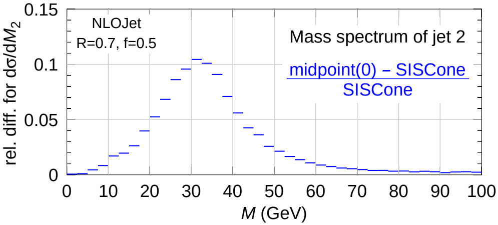
- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$

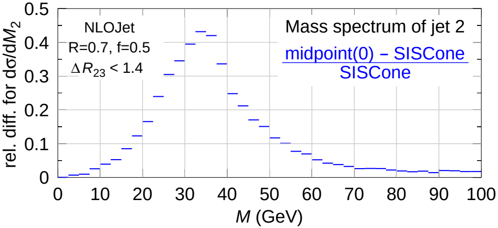
  e.g. for jets from common decay chain

In complex events, IR safety matters

IR safety & multi-jet observables

Look at jet masses in multijet events. NB: Jet masses reconstruct boosted $W/Z/H/top$ in BSM searches





Select 3-jet events

$$p_{t1,2,3} > \{120, 60, 20\} \text{ GeV},$$

Calculate LO jet-mass spectrum for jet 2, compare midpoint with SISCone.

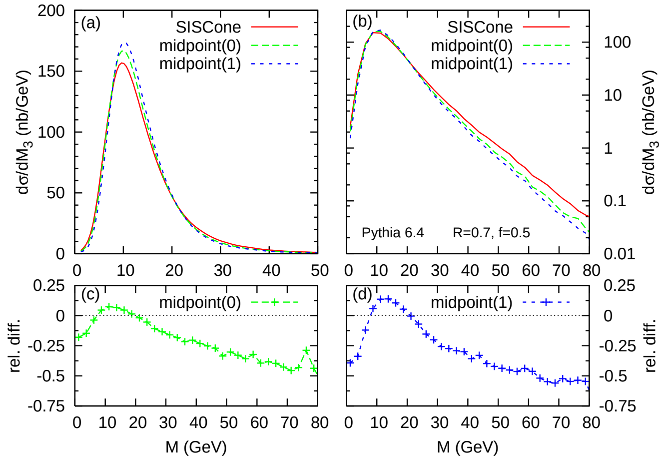
- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$
  e.g. for jets from common decay chain

In complex events, IR safety matters

Multi-jet observables: after showering

Showering puts in many extra seeds: missing stable cones (in midpoint) should be less important?

Look at 3rd jet mass distribution (no ΔR_{23} cut):



Missing stable cones → 50% effects even after showering

A full set of IRC-safe jet algorithms

Generalise inclusive-type sequential recombination with

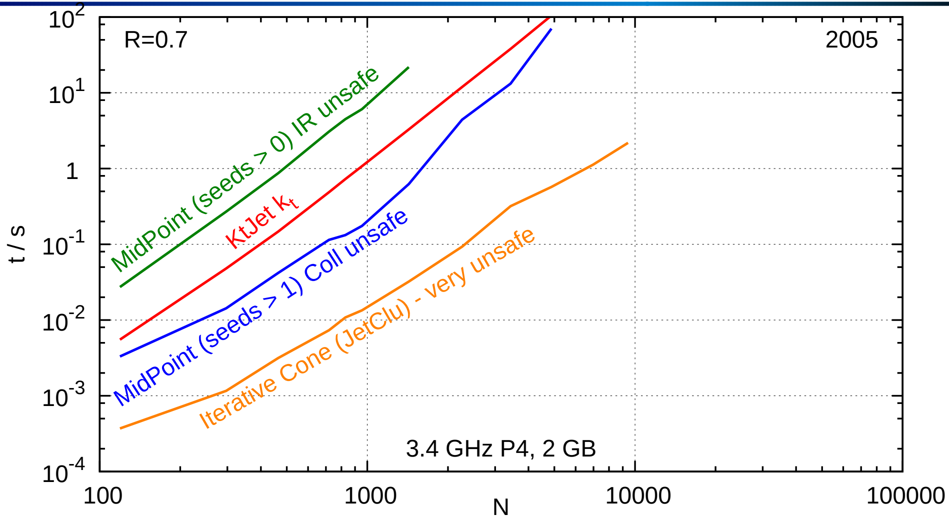
$$d_{ij} = \min(k_{ti}^{2\mathbf{p}}, k_{tj}^{2\mathbf{p}})\Delta R_{ij}^2/R^2 \qquad d_{iB} = k_{ti}^{2\mathbf{p}}$$

	Alg. name	Comment	time
$p = 1$	k_t CDOSTW '91-93; ES '93	Hierarchical in rel. k_t	$N \ln N$ exp.
$p = 0$	Cambridge/Aachen Dok, Leder, Moretti, Webber '97 Wengler, Wobisch '98	Hierarchical in angle Scan multiple R at once $\leftrightarrow$ QCD angular orderin	$N \ln N$
$p = -1$	anti-k_t Cacciari, GPS, Soyez '08 $\sim$ reverse-k_t Delsart, Loch et al.	Hierarchy meaningless. Behaves like IC-PR	$N^{3/2}$
SC-SM	SISCone GPS Soyez '07 + Tevatron run II '00	Replacement for IC-SM notably "MidPoint" cones	$N^2 \ln N$ exp.

One could invent/try others (e.g. OJF, etc.). Our [Paris+BNL] philosophy: 4 algs is enough of a basis to develop first physics understanding.

We already have far more than can be shown here

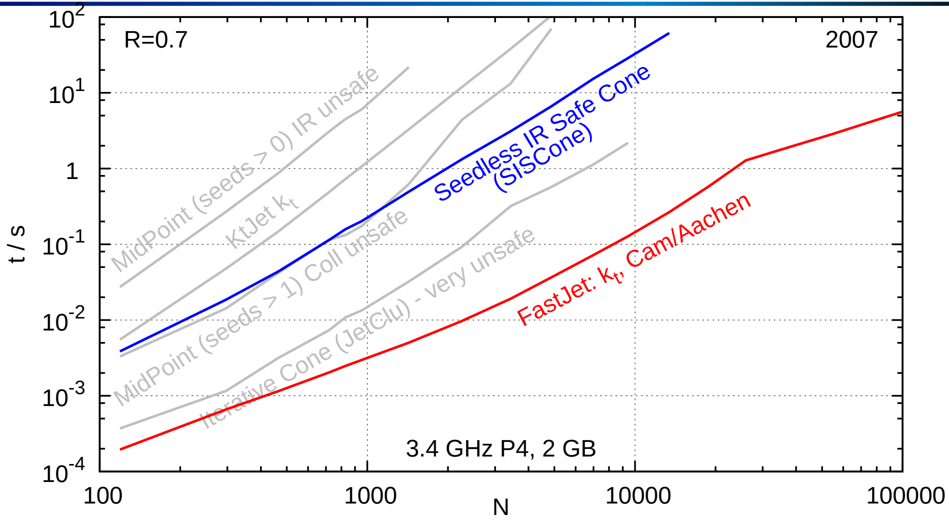
Status in 2005



Single package, **FastJet**, to access all developments, natively (k_t, Cam/Aachen) or as plugins (SISCone): Cacciari, GPS & Soyez '05–07

http://www.lpthe.jussieu.fr/~salam/fastjet/

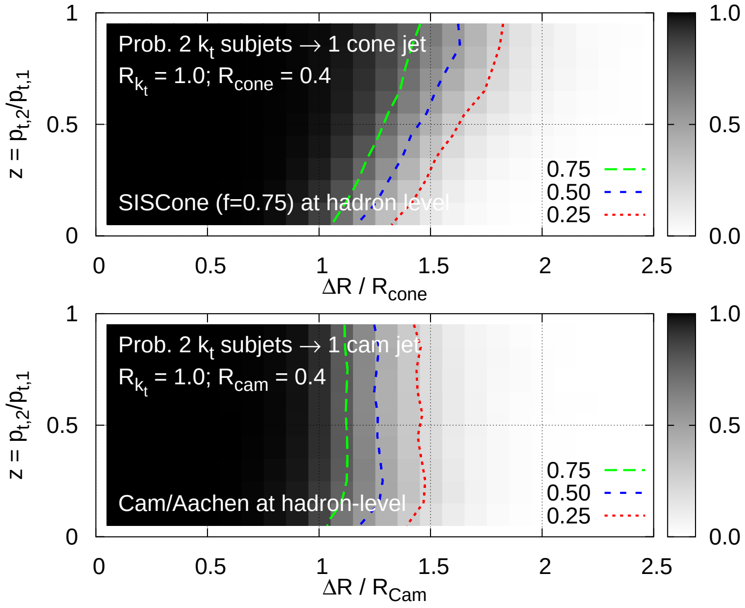
Status in 2007



Single package, **FastJet**, to access all developments, natively (k_t, Cam/Aachen) or as plugins (SISCone): Cacciari, GPS & Soyez '05–07

http://www.lpthe.jussieu.fr/~salam/fastjet/

Reach of jet algorithms



Herwig 6.510 + FastJet 2.1