A Practical Seedless Infrared Safe Cone Algorithm

Gavin Salam

LPTHE, Universities of Paris VI and VII and CNRS

work done in collaboration with Gregory Soyez

XLII Rencontres de Moriond — QCD and Hadronic interactions
17–24 March 2007

Two classes of jet algorithm

Sequential recombination	**Cone**
k_t, Jade, Cam/Aachen, ...	UA1, JetClu, Midpoint, ...
Bottom-up: Cluster 'closest' particles repeatedly until few left $\rightarrow$ jets.	Top-down: Find coarse regions of energy flow (cones), and call them jets.
Works because of mapping: *closeness* $\Leftrightarrow$ *QCD divergence*	Works because *QCD only modifies energy flow on small scales*
Loved by e^+e^-, *ep* and theorists	Loved by *pp* and few(er) theorists

Cone jet-algorithms will be everywhere at LHC
BUT: so far they don't meet the standards set out $>$ 15 years ago...

Two classes of jet algorithm

Sequential recombination	**Cone**
k_t, Jade, Cam/Aachen, ...	UA1, JetClu, Midpoint, ...
Bottom-up: Cluster 'closest' particles repeatedly until few left $\rightarrow$ jets.	Top-down: Find coarse regions of energy flow (cones), and call them jets.
Works because of mapping: *closeness* $\Leftrightarrow$ *QCD divergence*	Works because *QCD only modifies energy flow on small scales*
Loved by e^+e^-, *ep* and theorists	Loved by *pp* and few(er) theorists

Cone jet-algorithms will be everywhere at LHC
BUT: so far they don't meet the standards set out $>$ 15 years ago...

1990 “standards”

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.

With current cone algs., you have a choice: forego 1, 2 or 4

This talk will explain the problem and show how to solve it

1990 "standards"

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.

With current cone algs., you have a choice: forego 1, 2 or 4

This talk will explain the problem and show how to solve it

1990 “standards”

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.

With current cone algs., you have a choice: forego 1, 2 or 4

This talk will explain the problem and show how to solve it

Cone basics

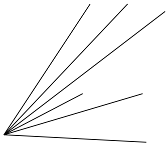
Modern cone algs have two main steps:

- 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



Cone basics

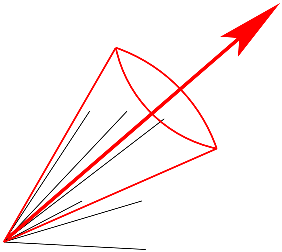
Modern cone algs have two main steps:

- 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



Cone basics

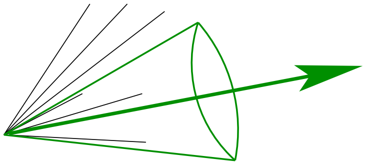
Modern cone algs have two main steps:

- 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



Cone basics

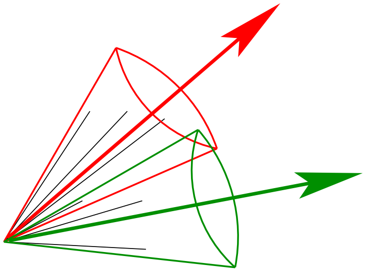
Modern cone algs have two main steps:

- 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



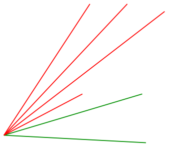
Cone basics

Modern 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



Cone basics

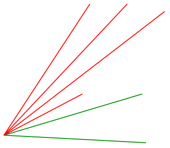
Modern cone algs have two main steps:

- 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



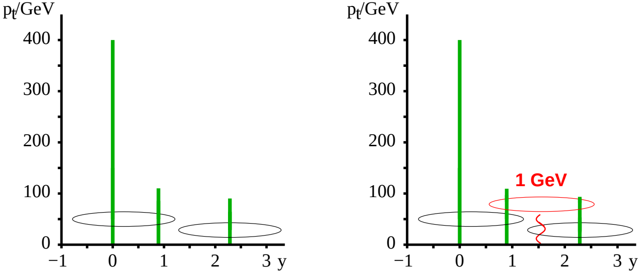
Qu: How do you find the stable cones?

All experiments use iterative methods:

- use each particle as a starting direction for cone; use sum of contents as new starting direction; repeat.
- use additional 'midpoint' starting points between pairs of initial stable cones.

'Midpoint' algorithm

Midpoint IR problem



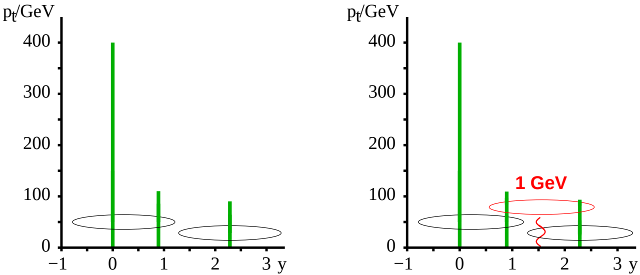
Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

Midpoint cone alg. misses some stable cones; extra soft particle → extra starting point → extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR problem



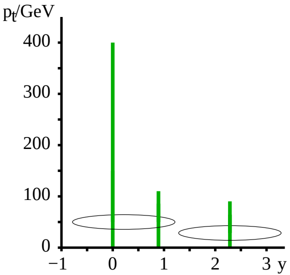
Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

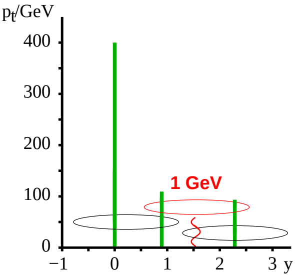
Midpoint cone alg. misses some stable cones; extra soft particle → extra starting point → extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR problem





Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

Midpoint cone alg. misses some stable cones; extra soft particle → extra starting point → extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR unsafety? Who cares?

Midpoint was supposed to solve *just this type of problem.* But worked only at lowest order.

IR/Collinear unsafety is a serious problem!

- Invalidates theorems that ensure finiteness of perturbative QCD

  Cancellation of real & virtual divergences

- Destroys usefulness of (intuitive) partonic picture

  you cannot think in terms of hard partons if adding a 1 GeV gluon changes 100 GeV jets

- 'Pragmatically:' limits accuracy to which it makes sense to calculate

Process	1st miss cones @	Last meaningful order
Inclusive jets	NNLO	NLO [NNLO being worked on]
$W/Z + 1$ jet	NNLO	NLO
3 jets	NLO	LO [NLO in nlojet++]
$W/Z + 2$ jets	NLO	LO [NLO in MCFM]
jet masses in $2j + X$	LO	**none**

Midpoint IR unsafety? Who cares?

Midpoint was supposed to solve *just this type of problem.* But worked only at lowest order.

IR/Collinear unsafety is a serious problem!

- Invalidates theorems that ensure finiteness of perturbative QCD

  Cancellation of real & virtual divergences

- Destroys usefulness of (intuitive) partonic picture

  you cannot think in terms of hard partons if adding a 1 GeV gluon changes 100 GeV jets

- 'Pragmatically:' limits accuracy to which it makes sense to calculate

Process	1st miss cones @	Last meaningful order
Inclusive jets	NNLO	NLO [NNLO being worked on]
$W/Z + 1$ jet	NNLO	NLO
3 jets	NLO	LO [NLO in `nlojet++`]
$W/Z + 2$ jets	NLO	LO [NLO in `MCFM`]
jet masses in $2j + X$	LO	**none**

Seedless cone algorithms

Rather than define the cone alg. through the *procedure* you use to find cones, define it by the *result you want:*

A cone algorithm should find **all** stable cones

First advocated: Kidonakis, Oderda & Sterman '97

Guarantees IR safety of the set of stable cones

Only issue: you still need to find the stable cones in practice.

One known exact approach:

- Take each possible subset of particles and see if it forms a stable cone.

  Tevatron Run II workshop, '00 (for fixed-order calcs.)

- There are 2^N subsets for N particles. Computing time $\sim \mathbf{N2^N}$.

  10^{17} years for an event with 100 particles

Seedless cone algorithms

Rather than define the cone alg. through the *procedure* you use to find cones, define it by the *result you want:*

A cone algorithm should find **all** stable cones

First advocated: Kidonakis, Oderda & Sterman '97
Guarantees IR safety of the set of stable cones

Only issue: you still need to find the stable cones in practice.

One known exact approach:

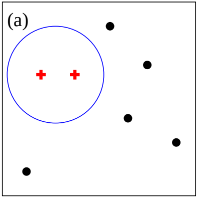
- Take each possible subset of particles and see if it forms a stable cone.
  Tevatron Run II workshop, '00 (for fixed-order calcs.)
- There are 2^N subsets for N particles. Computing time $\sim \mathbf{N2^N}$.
  10^{17} years for an event with 100 particles

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

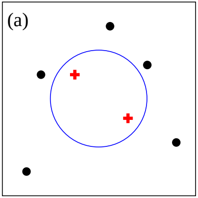
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

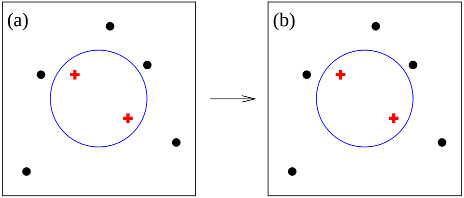
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

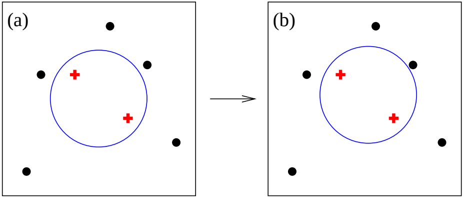
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

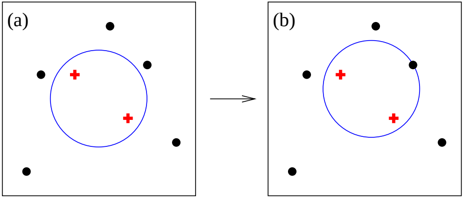
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

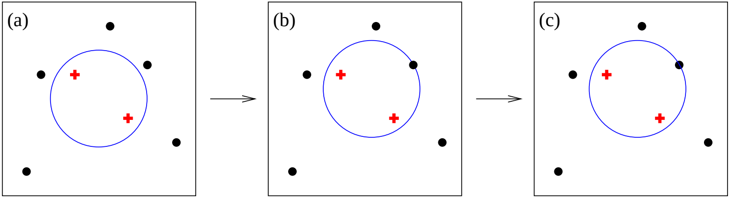
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

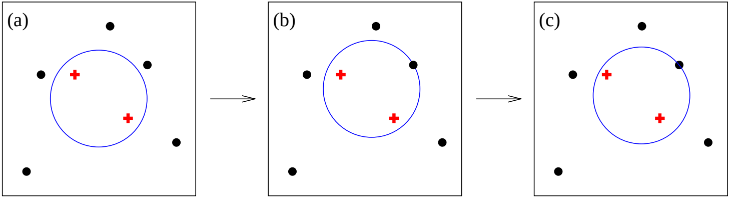
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

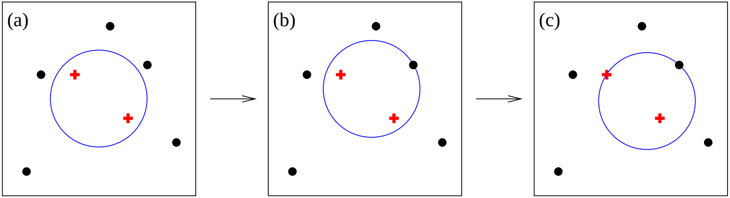
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

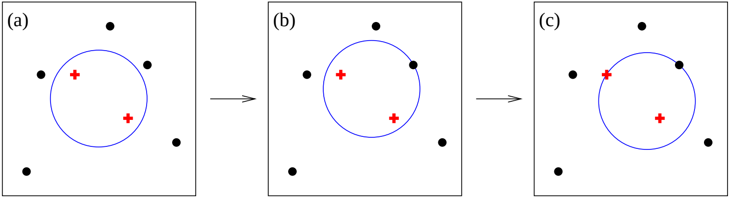
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

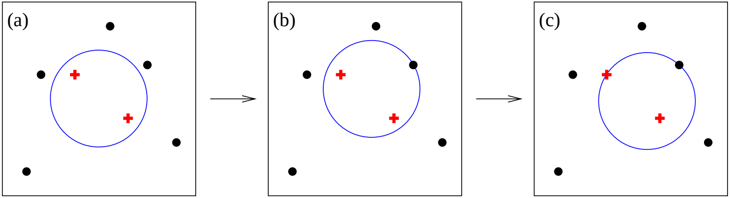
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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.

Polynomial time recipe for finding all distinct enclosures:

- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

A Seedless Infrared Safe Cone: SISCone

Naive implementation of this idea would run in N^3 time.

N^2 pairs of points, pay N for each pair to check stability

N^3 is also time taken by midpoint codes (smaller coeff.)

With some thought, this reduces to $\mathbf{N^2 \ln N}$ time.

Traversal order, stability check

checkxor

- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

  Collinear unsafe

NB kt & Cam/Aachen (seq. recomb.) algs are much faster

A Seedless Infrared Safe Cone: SISCone

Naive implementation of this idea would run in N^3 time.

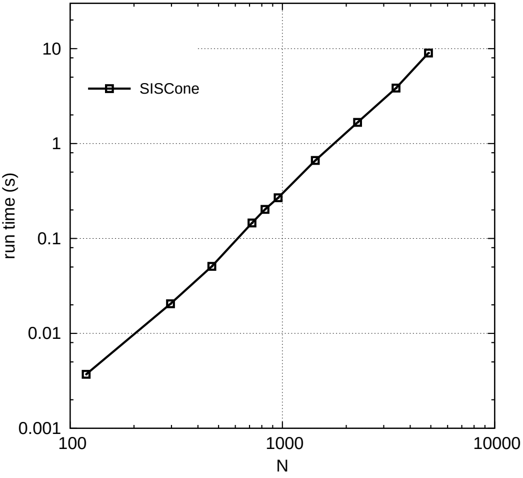
N^2 pairs of points, pay N for each pair to check stability

N^3 is also time taken by midpoint codes (smaller coeff.)

With some thought, this reduces to **$N^2 \ln N$** time.

Traversal order, stability check

checkxor



- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds $>$ 1 GeV

  Collinear unsafe

NB kt & Cam/Aachen (seq. recomb.) algs are much faster

A Seedless Infrared Safe Cone: SISCone

Naive implementation of this idea would run in N^3 time.

N^2 pairs of points, pay N for each pair to check stability

N^3 is also time taken by midpoint codes (smaller coeff.)

With some thought, this reduces to **$N^2 \ln N$** time.

Traversal order, stability check

checkxor

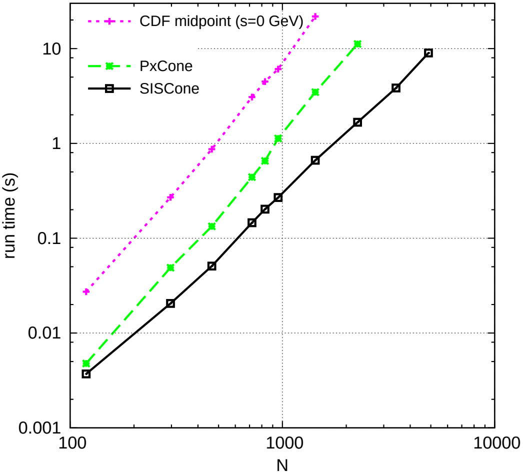
- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

  Collinear unsafe

NB kt & Cam/Aachen (seq. recomb.) algs are much faster



A Seedless Infrared Safe Cone: SISCone

Naive implementation of this idea would run in N^3 time.

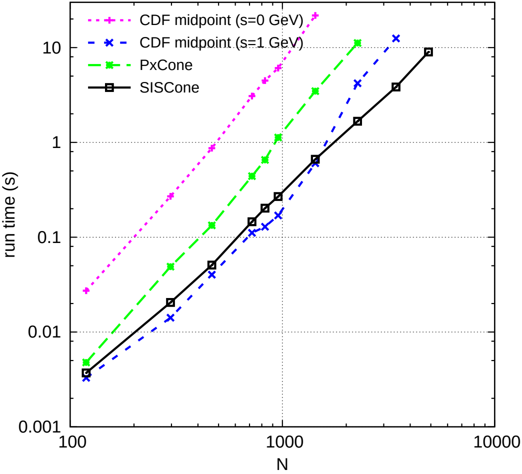
N^2 pairs of points, pay N for each pair to check stability

N^3 is also time taken by midpoint codes (smaller coeff.)

With some thought, this reduces to **$N^2 \ln N$** time.

Traversal order, stability check

checkxor



- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

  Collinear unsafe

NB kt & Cam/Aachen (seq. recomb.) algs are much faster

A Seedless Infrared Safe Cone: SISCone

Naive implementation of this idea would run in N^3 time.

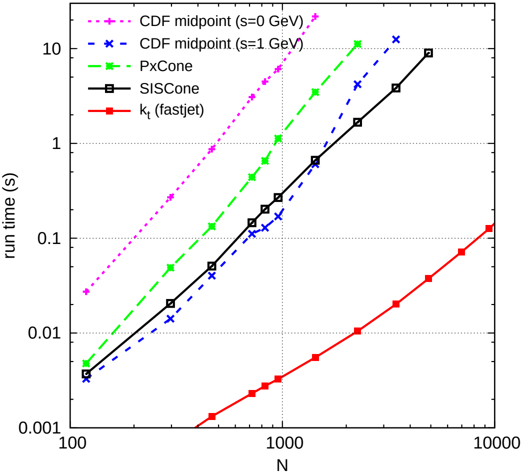
N^2 pairs of points, pay N for each pair to check stability

N^3 is also time taken by midpoint codes (smaller coeff.)

With some thought, this reduces to **$N^2 \ln N$** time.

Traversal order, stability check

checkxor



- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

  Collinear unsafe

NB kt & Cam/Aachen (seq. recomb.) algs are much faster

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 !

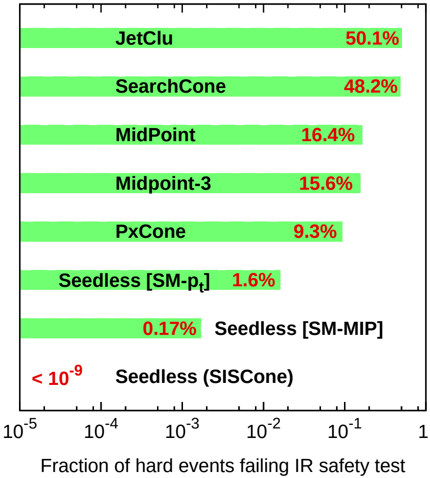
Be careful with split–merge too

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



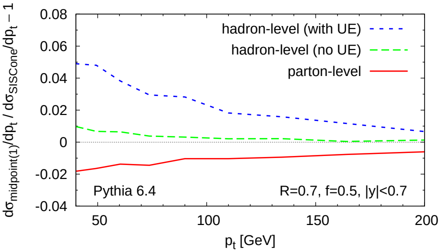
How much does IR safety *really* matter?

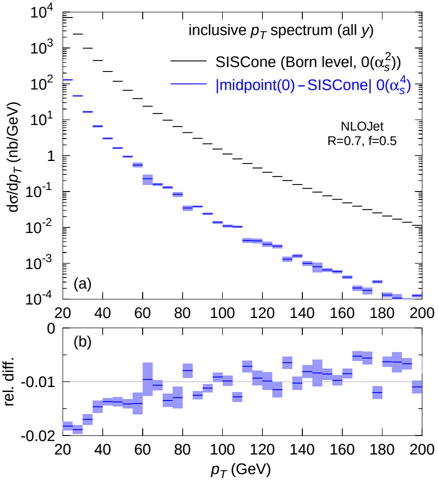
Compare midpoint and SISCone

Result depends on observable:

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

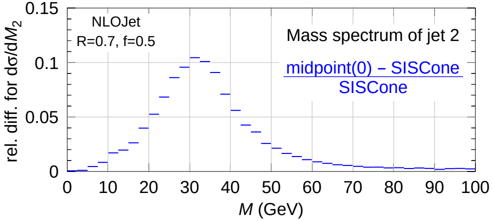
seedless reduces UE effect





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.

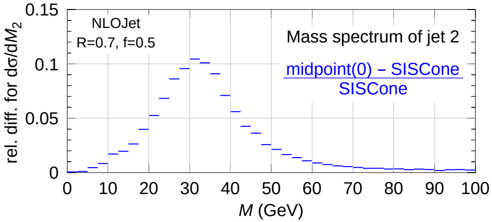
- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$

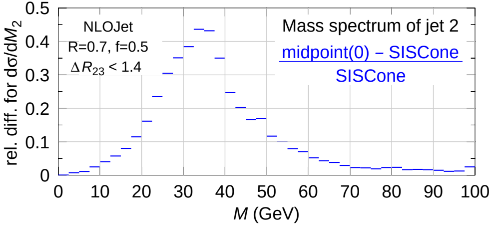
e.g. for jets from common decay chain

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 $\Delta R_{2,3} < 1.4$

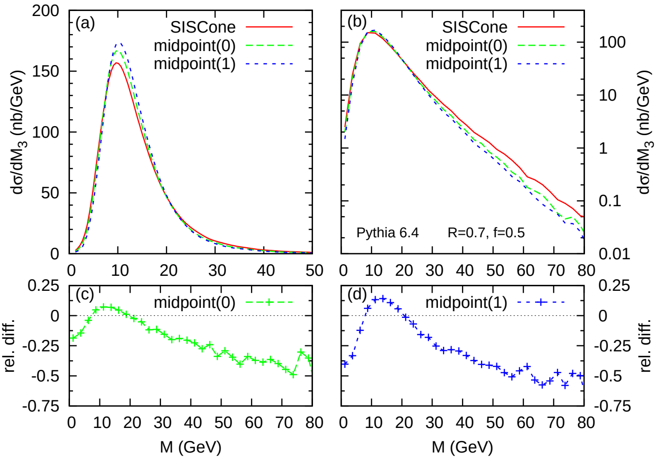
  e.g. for jets from common decay chain

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

Conclusions

- Currently-used cones have significant IR/Collinear safety problems
  Midpoint algorithm was an incomplete fix
- Cone algorithms *can be made* simultaneously IR/Coll safe and practical
  e.g. SISCone
- IR safety is not a luxury (effects most visible in complex events)
  Up to 40% effects; reduced UE sensitivity
- So if you use a cone algorithm, use a safe one

You can get SISCone from:

`http://projects.hepforge.org/siscone/` (standalone)

`http://www.lpthe.jussieu.fr/~salam/fastjet/` (FastJet plugin)

Extra Slides

Algorithm 1: SISCone 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 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 $N_{\rm pass}$ times.
7: Run a Tevatron Run-II type split–merge procedure, algorithm 3, on the full list of protojets, with overlap parameter f and transverse momentum threshold $p_{t,\min}$.

Algorithm 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

Algorithm 3: split–merge

1: **repeat**

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

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

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

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

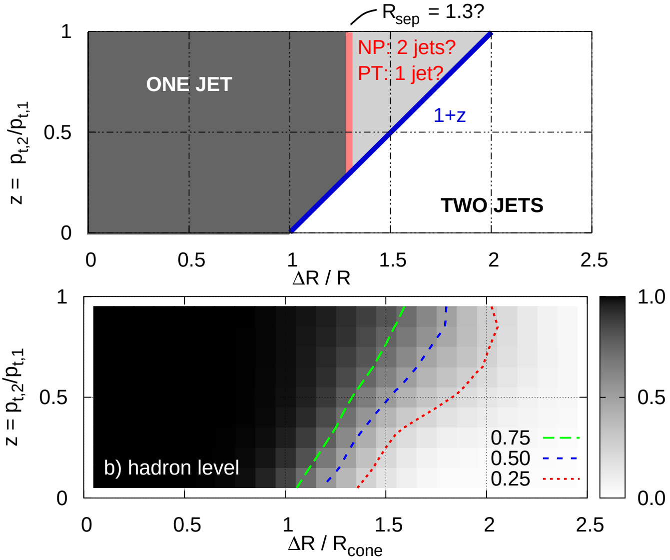
Add i to the list of final jets, and remove it from the list of protojets.

15: **end if**

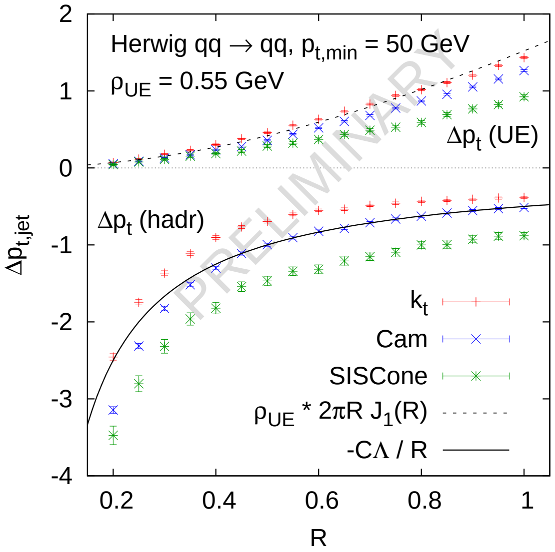
16: **until** no protojets are left.

R_{sep}

When do two partons (separated by ΔR, with $z = p_{t2}/p_{t1}$) recombine?



Non-pert. effects in 3 algorithms



Cacciari, Dasgupta, Magnea & GPS preliminary:

Single-gluon approx. for non-pert effects:

- Hadronisation:

$$\Delta p_{t,jet} \simeq -C_{F/A} \frac{0.35 \text{ GeV}}{R}$$

Coeff comes from e^+e^-

- U.E.

$$\Delta p_{t,jet} \simeq \rho_{UE} \cdot 2\pi R J_1(R)$$