Jets

Gavin Salam

LPTHE, CNRS and UPMC (Univ. Paris 6)

CTEQ MCNet school
Debrecen, Hungary, August 2008

Lecture 1 $\longrightarrow$ 2

In lecture 1, we saw

- sequential recombination (k_t, etc.) algorithms
- the first of a series of cone-algorithms, those with "progressive removal" (xC-PR)
- and ran into collinear safety issues (from ordering of "seeds" for cone direction)

Today

- see the other series of cone-algorithms (with split–merge, xC-SM)
- look more at the physics of jet algs.

Cone algorithms today

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

But cones come in many variants

Finding cones \ Processing	Progressive Removal	Split–Merge	Split–Drop
Seeded, Fixed (FC)	GetJet CellJet		
Seeded, Iterative (IC)	CMS Cone	JetClu (CDF)† ATLAS cone	
Seeded, It. + Midpoints (IC_{mp})		CDF MidPoint D0 Run II cone	PxCone
Seedless (SC)		SISCone	

† 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

Finding cones \ Processing	Progressive Removal	Split–Merge	Split–Drop
Seeded, Fixed (FC)	GetJet CellJet		
Seeded, Iterative (IC)	CMS Cone	JetClu (CDF)† ATLAS cone	
Seeded, It. + Midpoints (IC_{mp})		CDF MidPoint D0 Run II cone	PxCone
Seedless (SC)		SISCone	

† JetClu also has "ratcheting"

Cone algorithms today

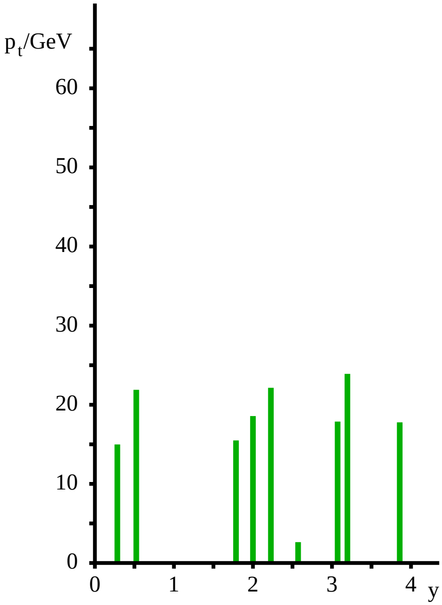
Unifying idea: momentum flow within a cone only marginally modified by QCD branching

But cones come in many variants

Finding cones \ Processing	Progressive Removal	Split–Merge	Split–Drop
Seeded, Fixed (FC)	GetJet CellJet		
Seeded, Iterative (IC)	CMS Cone	JetClu (CDF)† ATLAS cone	
Seeded, It. + Midpoints (IC_{mp})		CDF MidPoint D0 Run II cone	PxCone
Seedless (SC)		SISCone	

† JetClu also has "ratcheting"

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

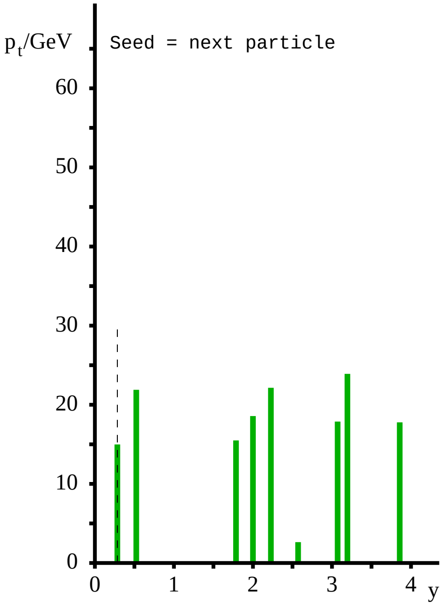
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a **split–merge** procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

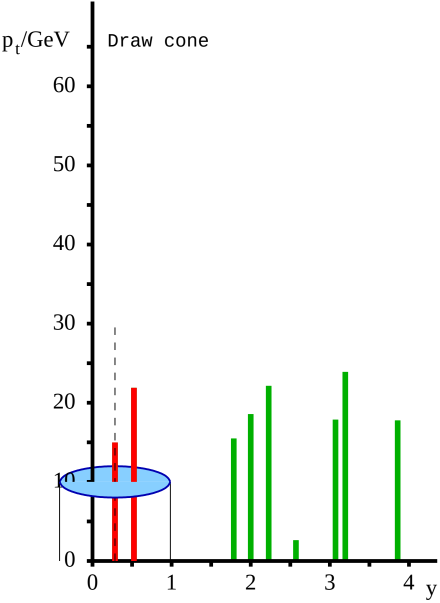
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)

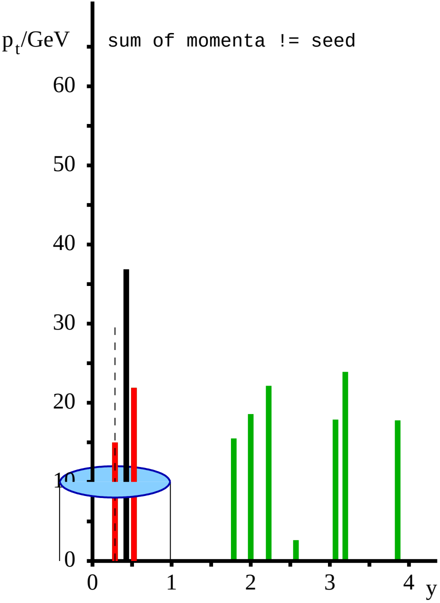


Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

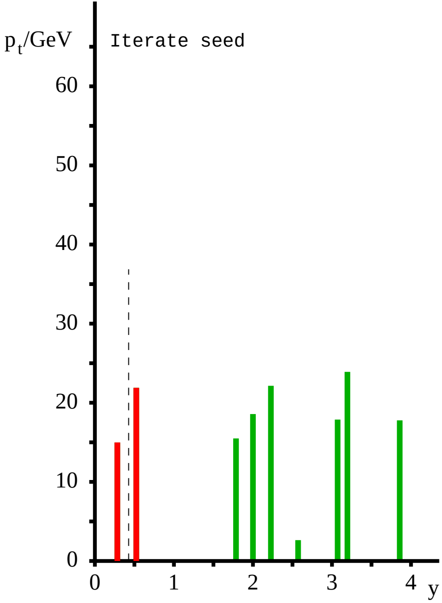
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet ≠ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

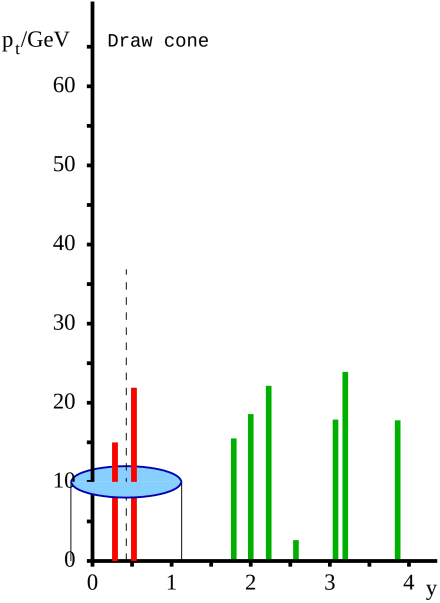
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

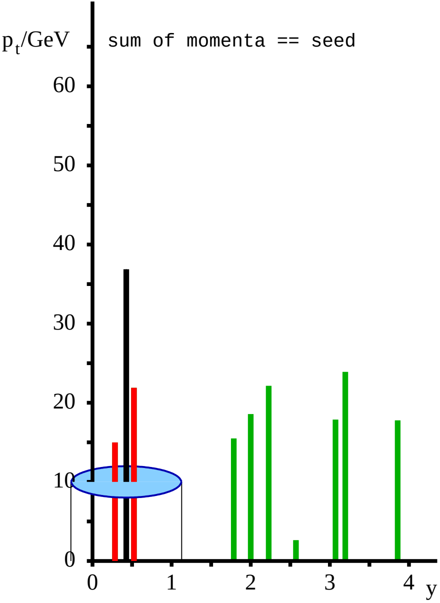
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

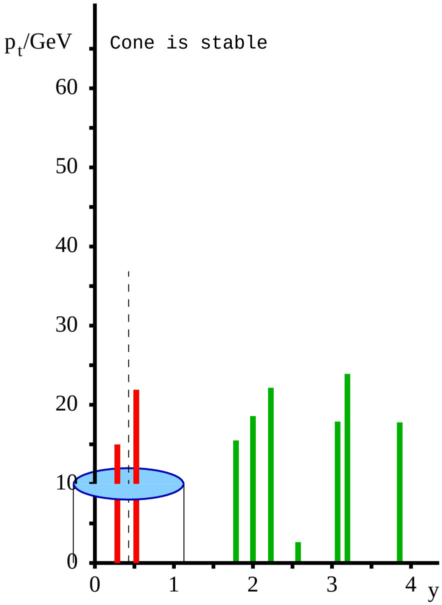
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

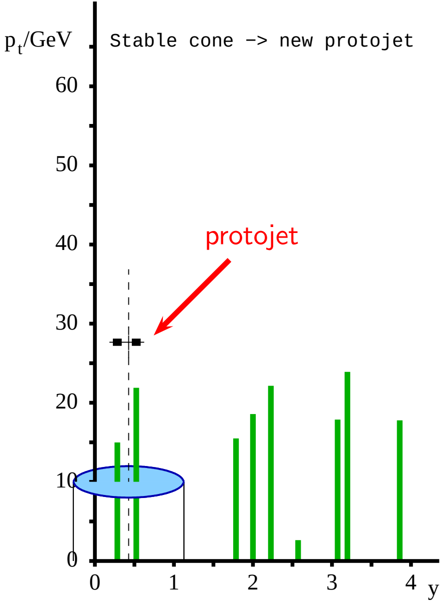
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet ≠ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

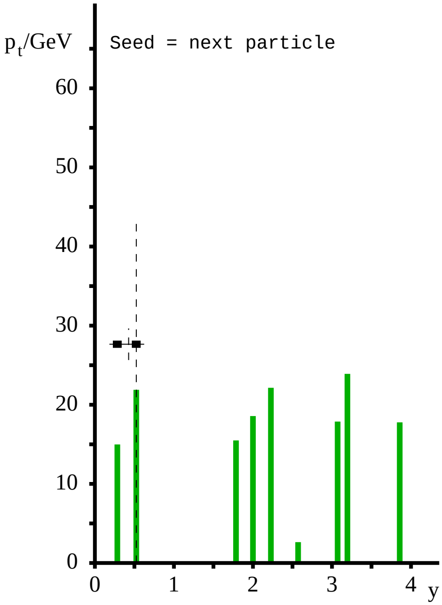
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

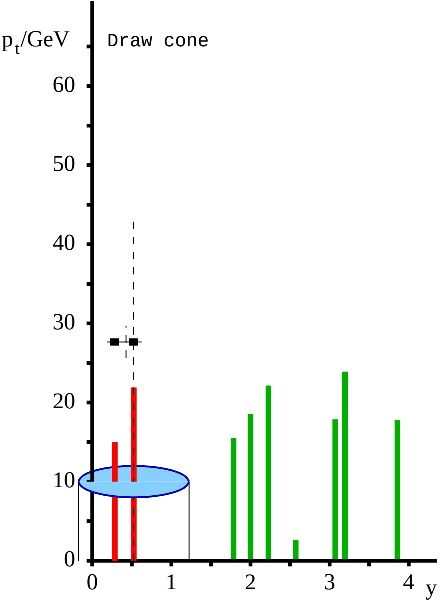
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)

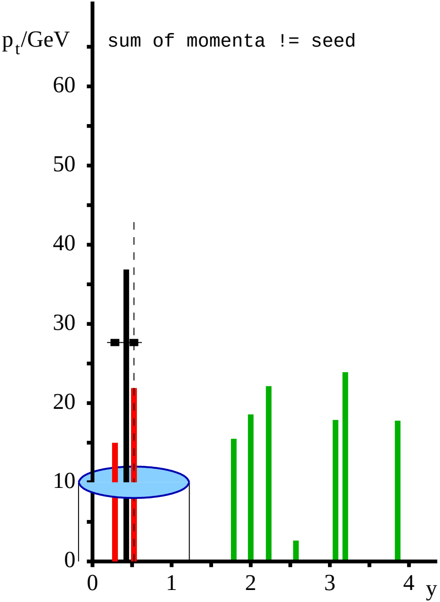


Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

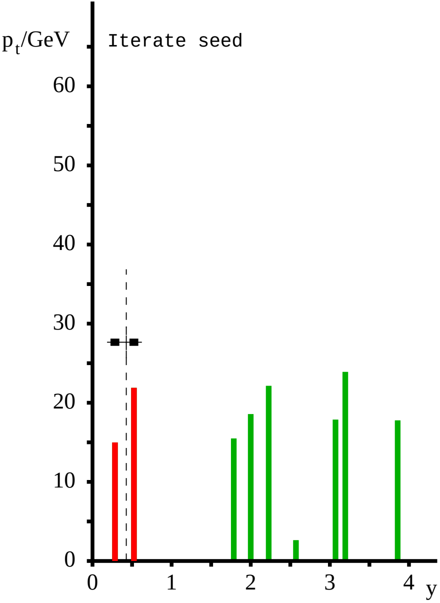
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

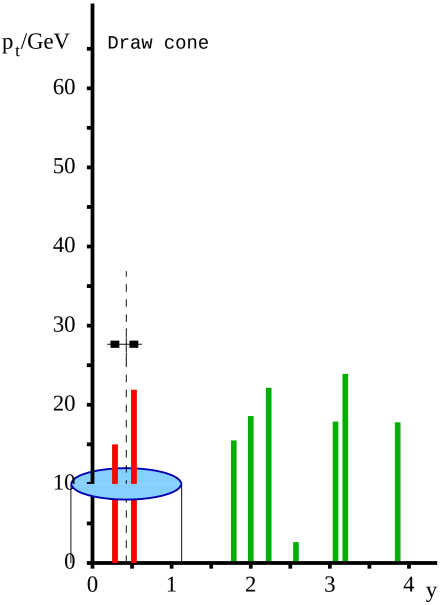
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

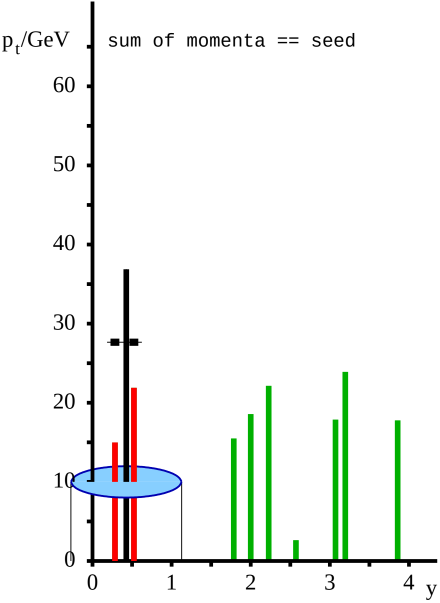
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

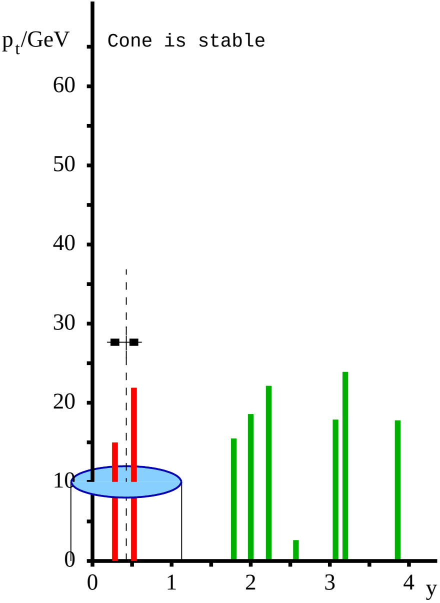
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

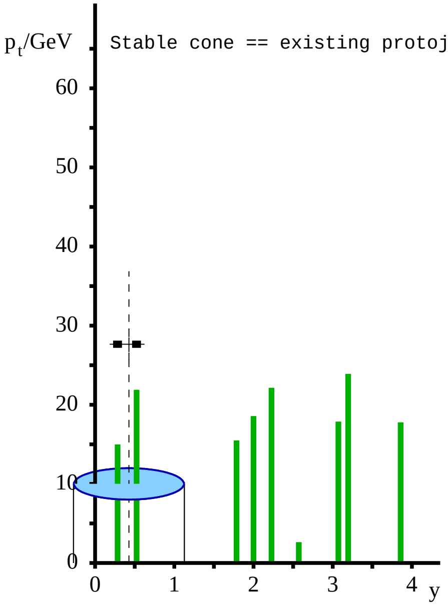
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet ≠ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

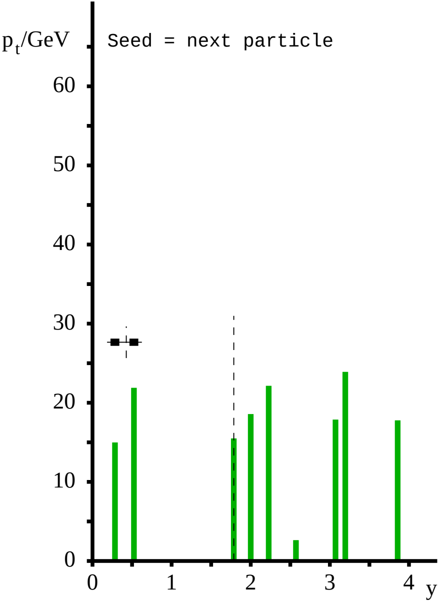
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

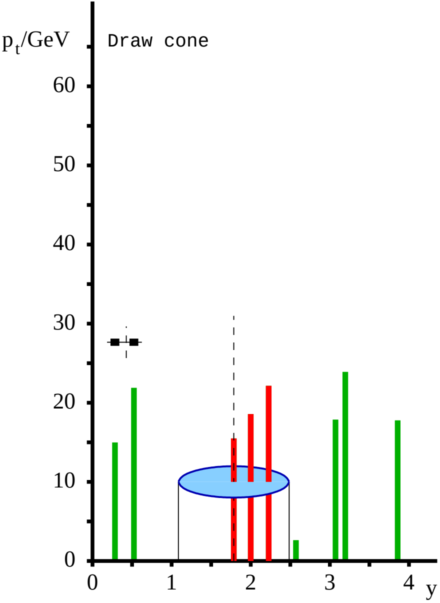
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

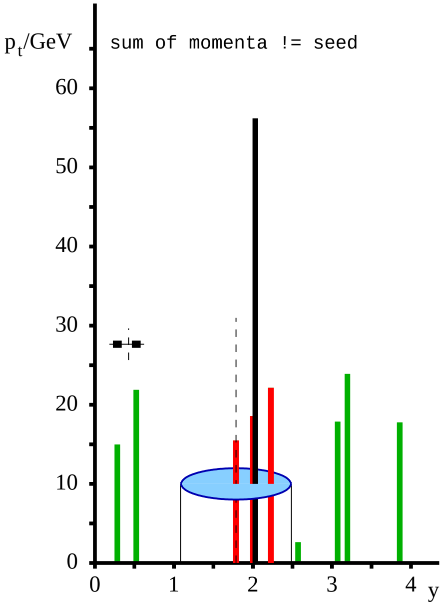
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

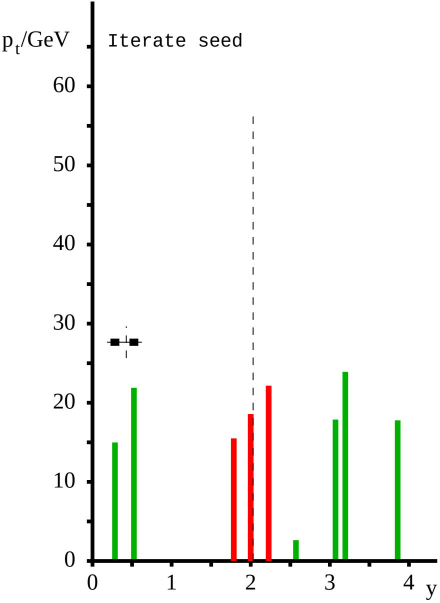
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet ≠ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

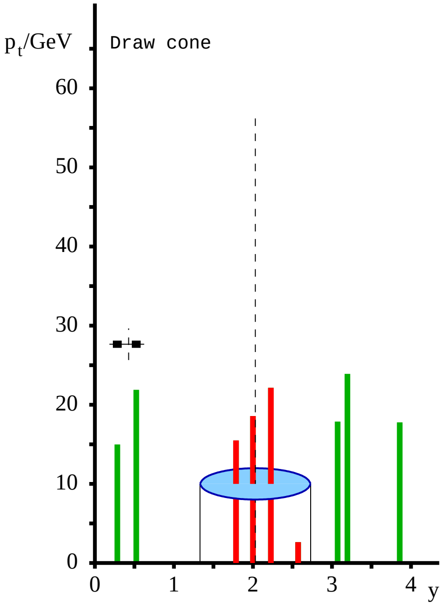
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps. Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

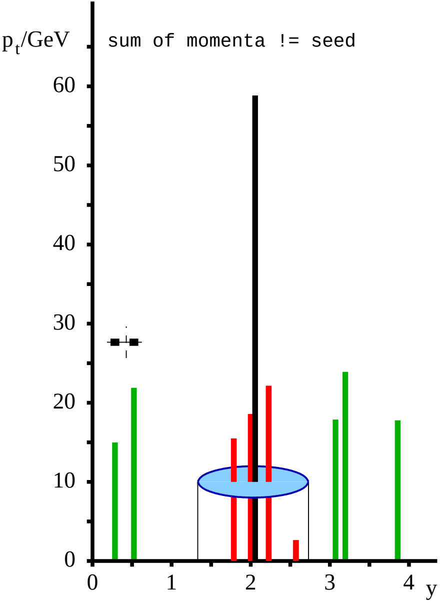
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

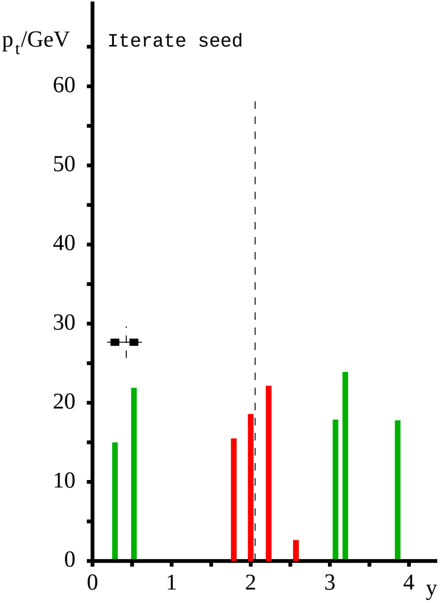
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

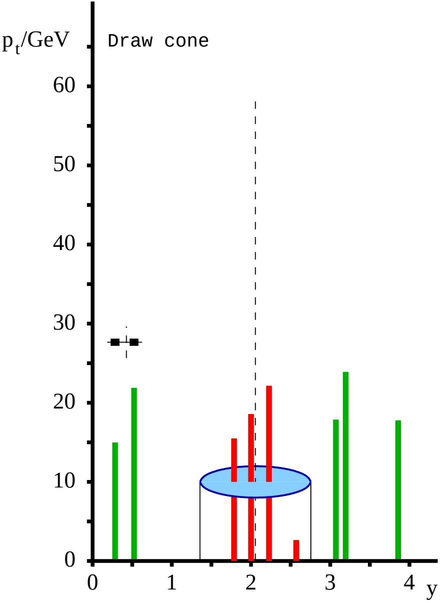
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

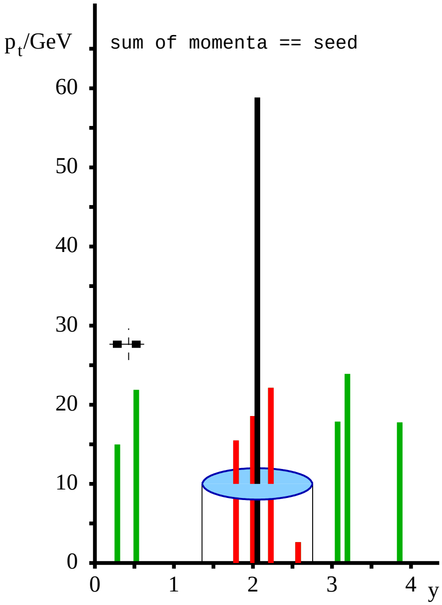
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet ≠ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

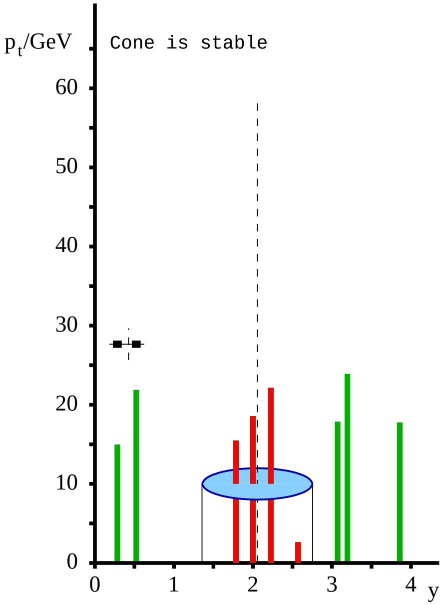
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets overlap. Certain particles appear in many protojets

protojet ≠ jet

Must resolve the overlaps.
Use a split–merge procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

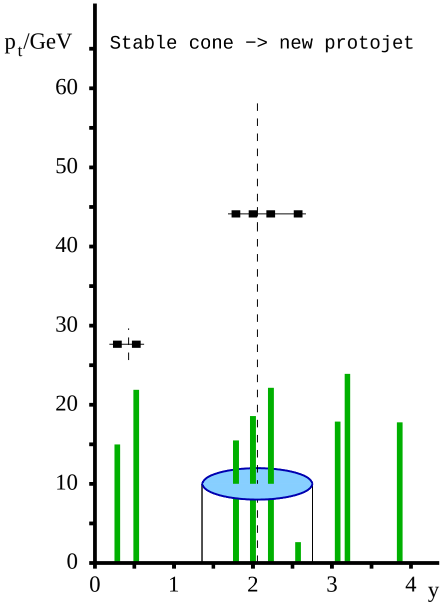
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a **split–merge** procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

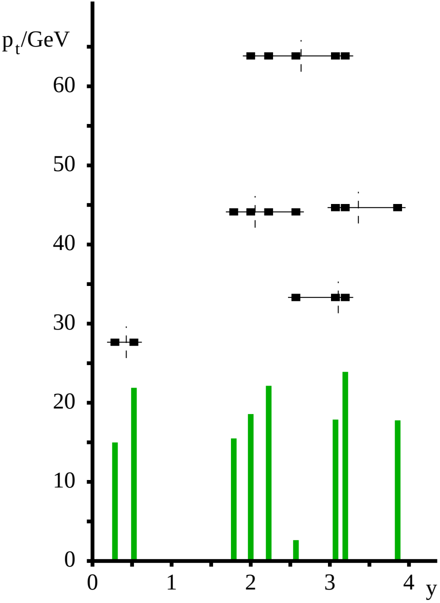
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet ≠ jet

Must resolve the overlaps.
Use a **split–merge** procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

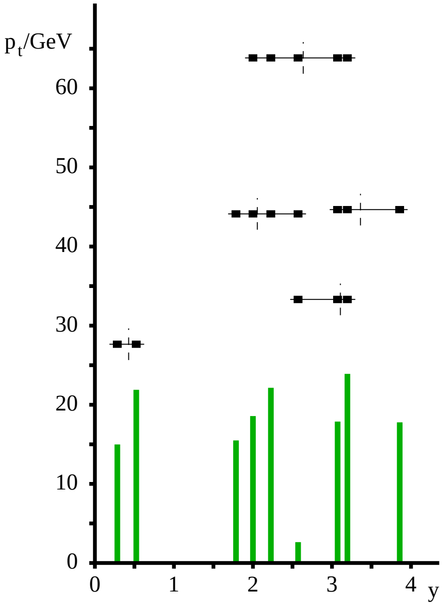
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a **split–merge** procedure.

It. Cone with Split–Merge (IC-SM)



Avoid ordering seeds (coll. unsafe)

CDF JetClu† & ATLAS cones

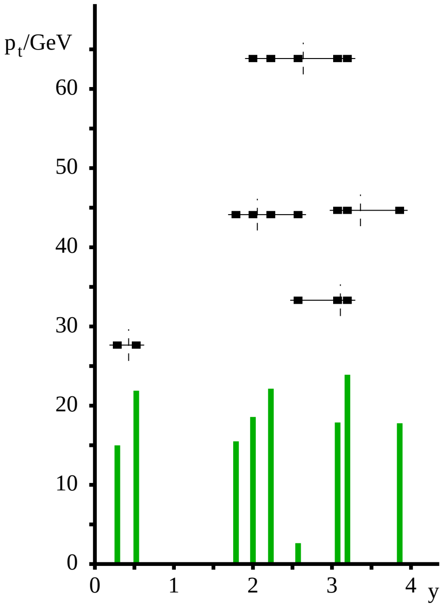
- use every particle as possible seed (no particular order)
- iterate until stable cone
- add the stable cone to the list of **protojets** unless it's already there
- until all seeds done

Note: protojets **overlap**. Certain particles appear in many protojets

protojet $\neq$ jet

Must resolve the overlaps.
Use a **split–merge** procedure.

IC-SM: split–merge part



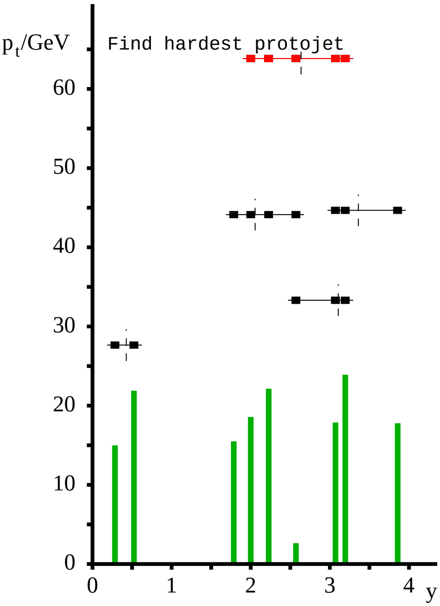
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



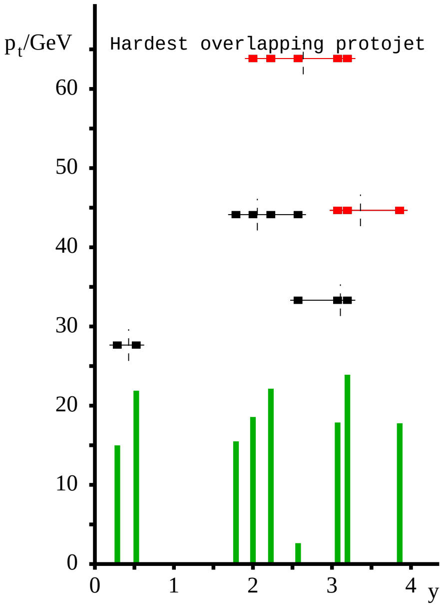
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\to$ jet.
- repeat...

IC-SM: split–merge part



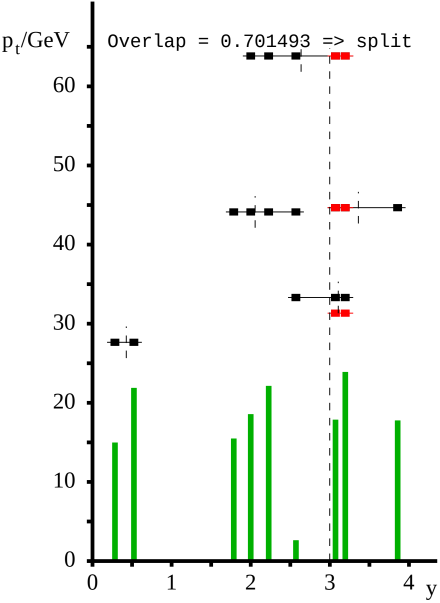
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



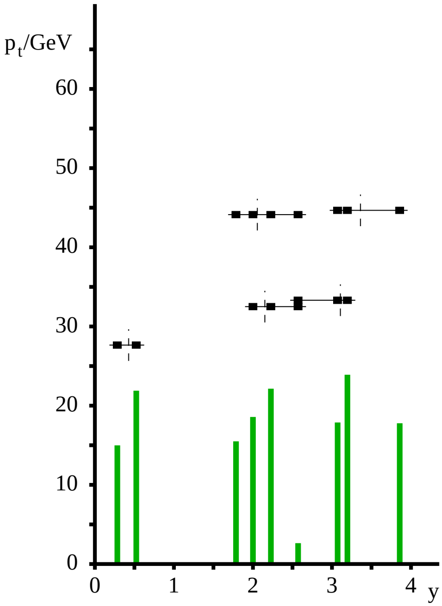
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat...

IC-SM: split–merge part



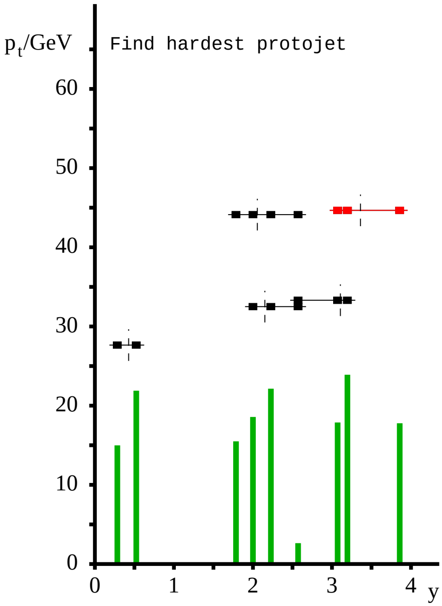
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



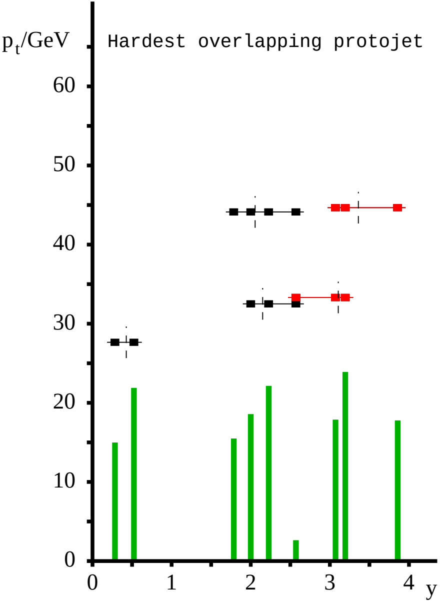
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



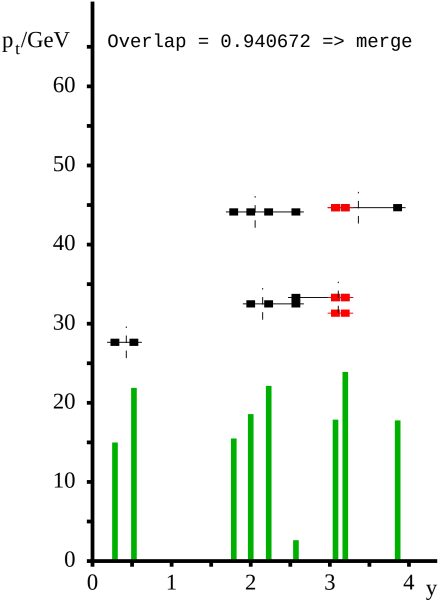
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap,
  $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



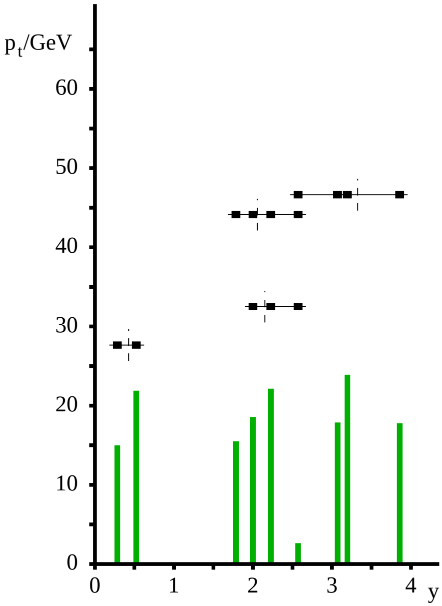
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



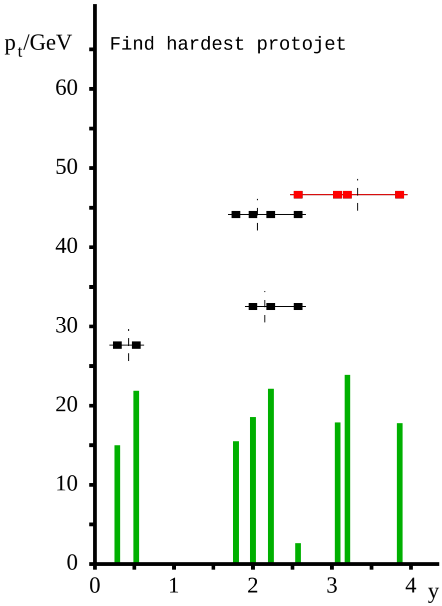
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



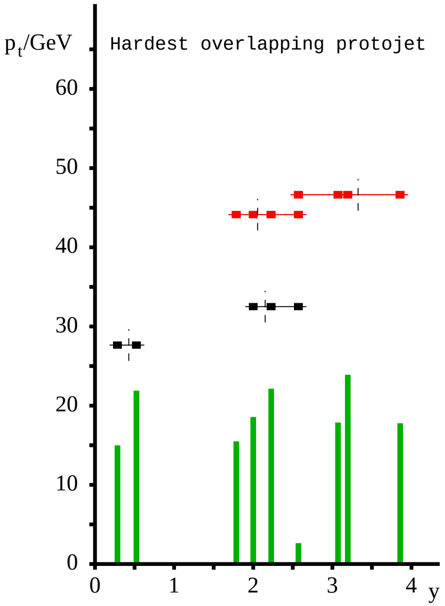
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



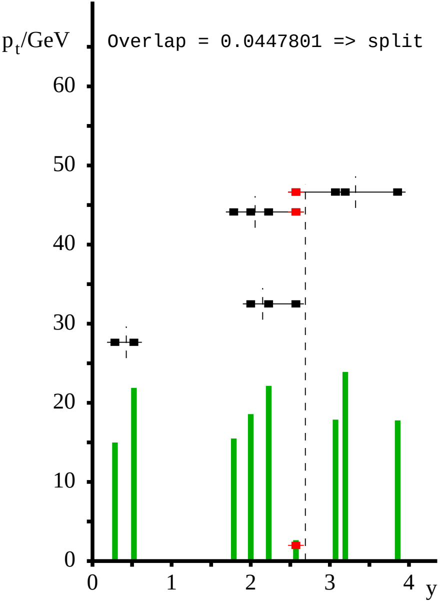
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\to$ jet.
- repeat...

IC-SM: split–merge part



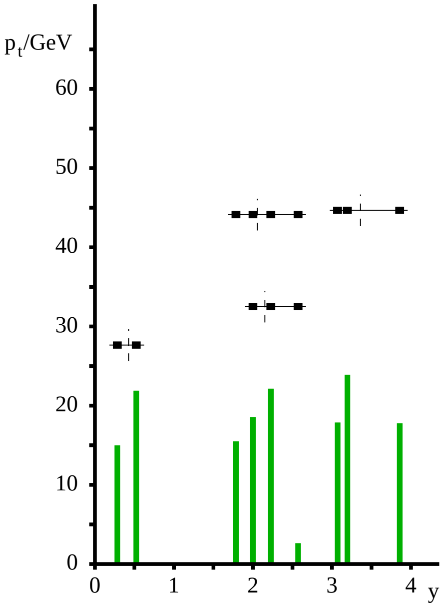
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



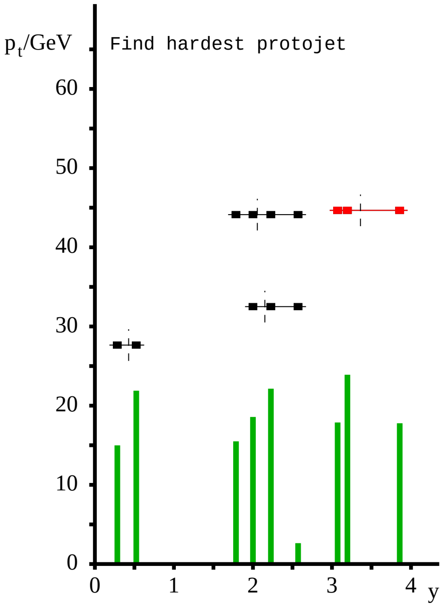
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



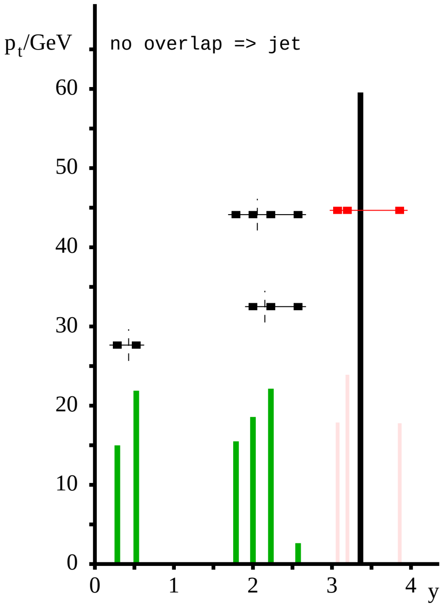
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat...

IC-SM: split–merge part



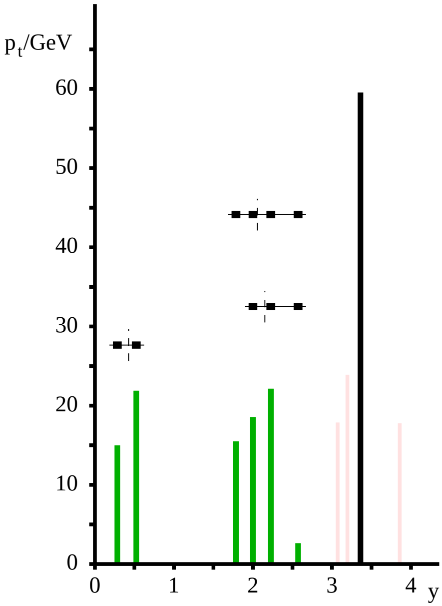
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ → jet.
- repeat...

IC-SM: split–merge part



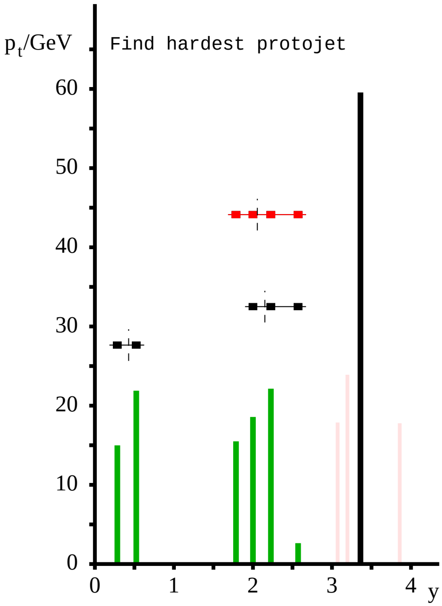
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\to$ jet.
- repeat...

IC-SM: split–merge part



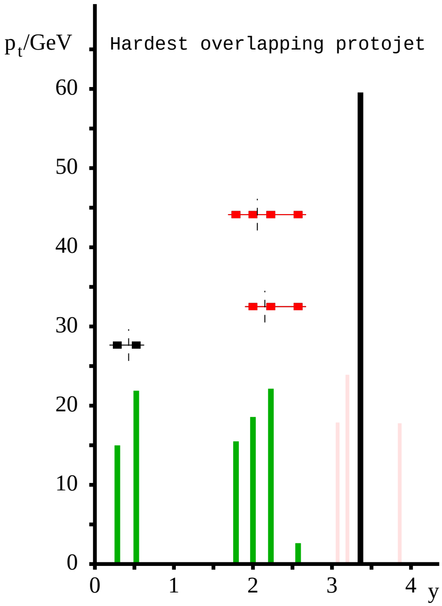
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\to$ jet.
- repeat...

IC-SM: split–merge part



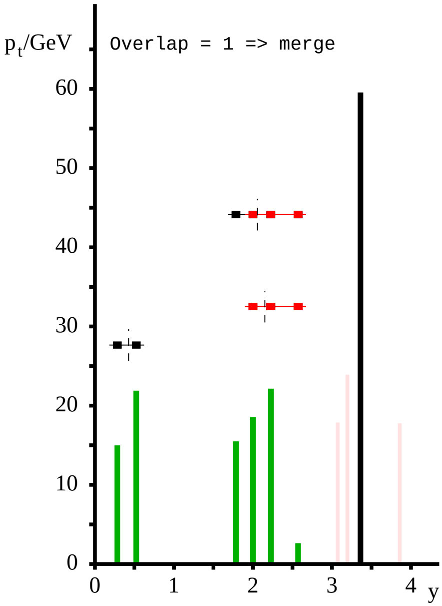
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



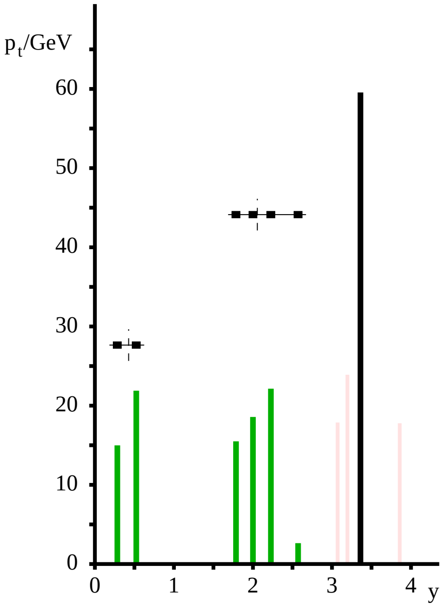
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\to$ jet.
- repeat. . .

IC-SM: split–merge part



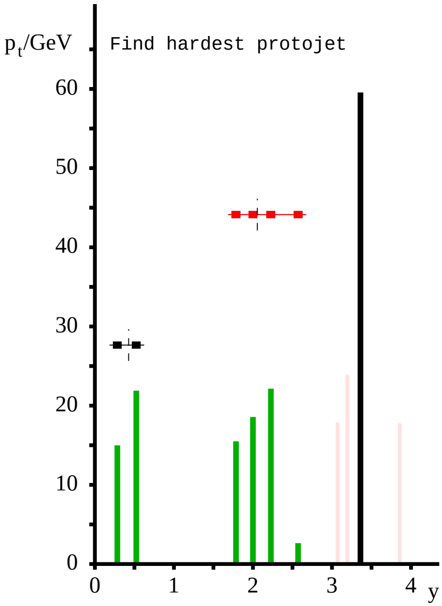
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat...

IC-SM: split–merge part



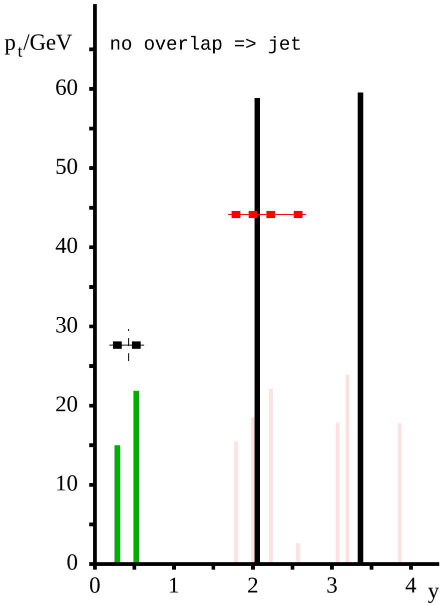
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

IC-SM: split–merge part



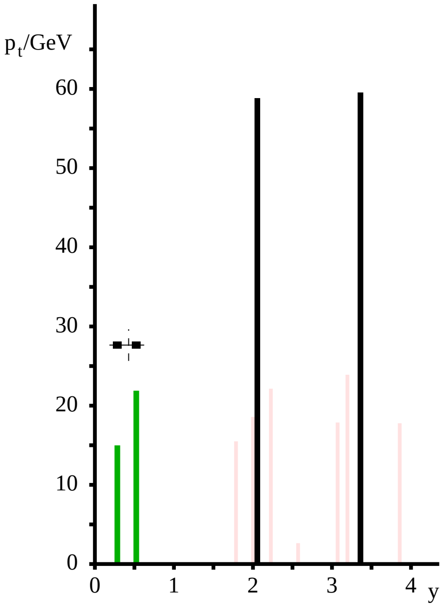
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\to$ jet.
- repeat...

IC-SM: split–merge part



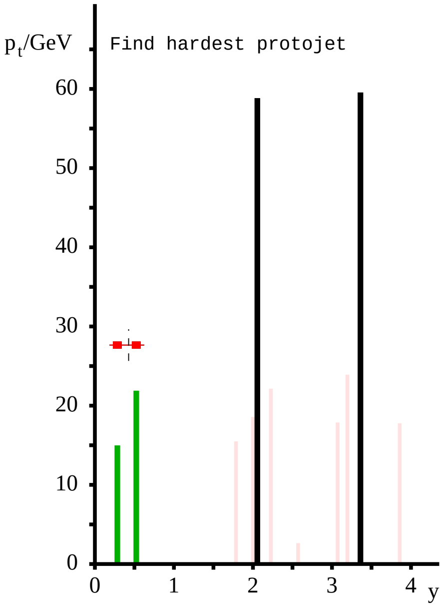
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat...

IC-SM: split–merge part



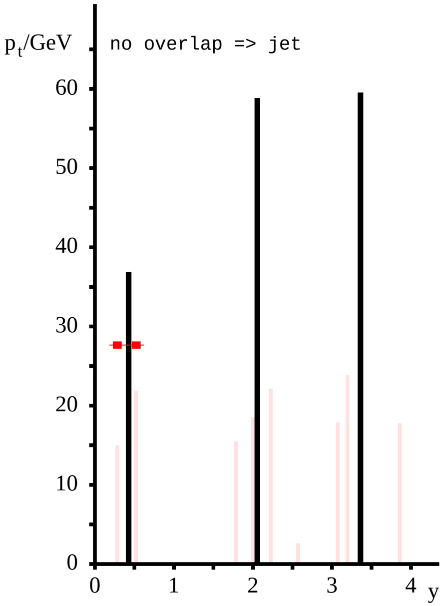
<u>SM in Tevatron Run II formulation</u>

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\to$ jet.
- repeat. . .

IC-SM: split–merge part



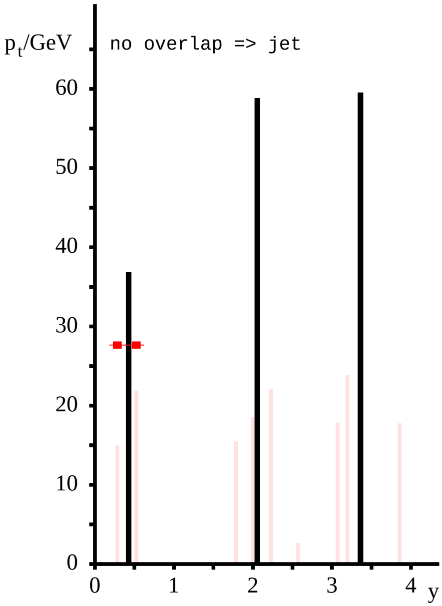
SM in Tevatron Run II formulation

but common to most xC-SM

Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared} / p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat...

IC-SM: split–merge part



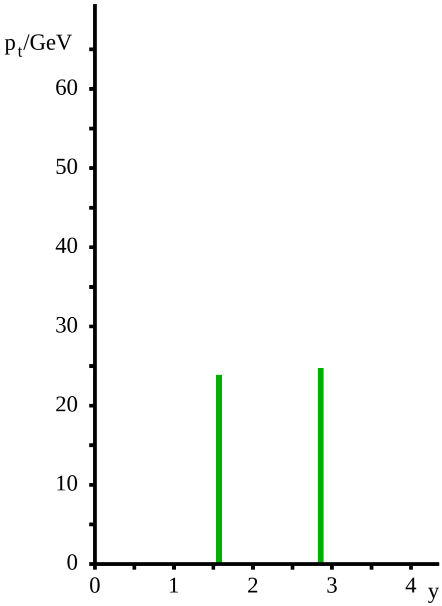
SM in Tevatron Run II formulation

but common to most xC-SM

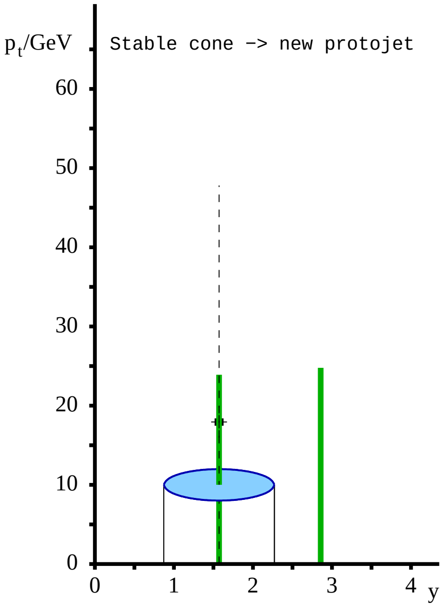
Introduce overlap threshold f

- Identify hardest protojet (PJ), p_1
- Find hardest PJ that overlaps with it, p_2
- Calculated overlap, $O = p_{t,shared}/p_{t,2}$
  - if $O < f$, split along axis at center of two PJs
  - if $O > f$ merge the two PJs
- If there is no overlap, PJ $\rightarrow$ jet.
- repeat. . .

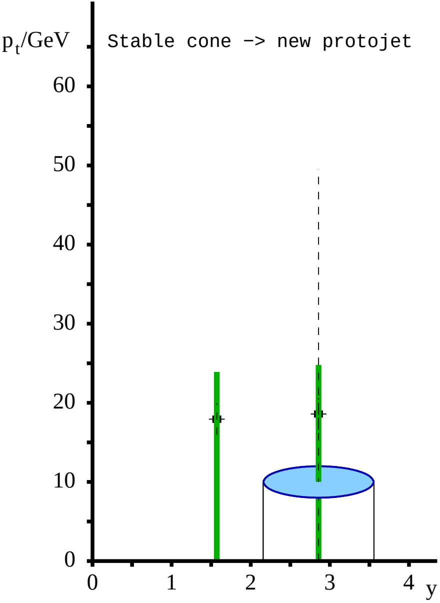
IC-SM: infrared safety



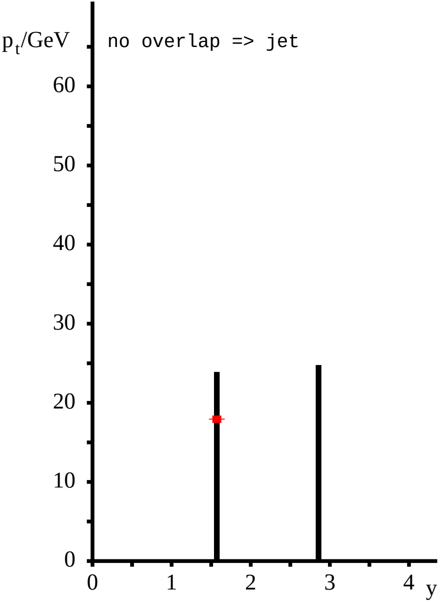
IC-SM: infrared safety



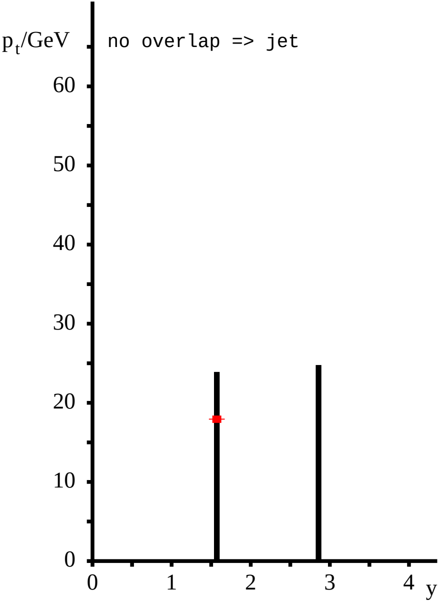
IC-SM: infrared safety

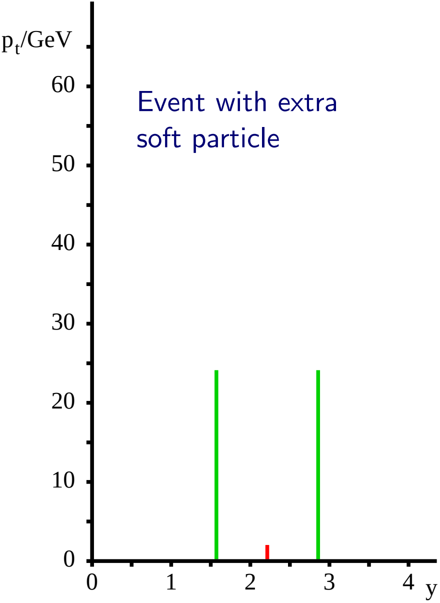


IC-SM: infrared safety

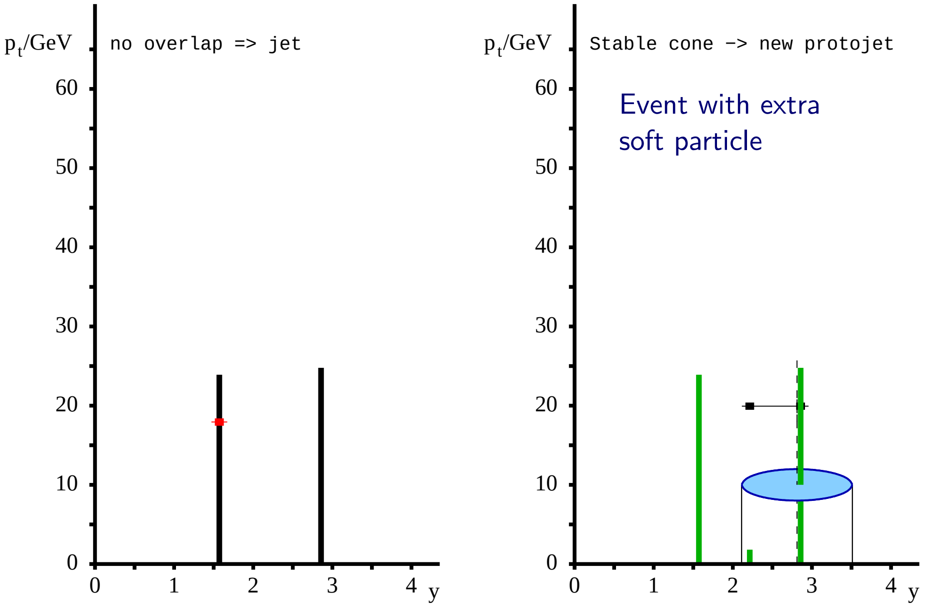


IC-SM: infrared safety

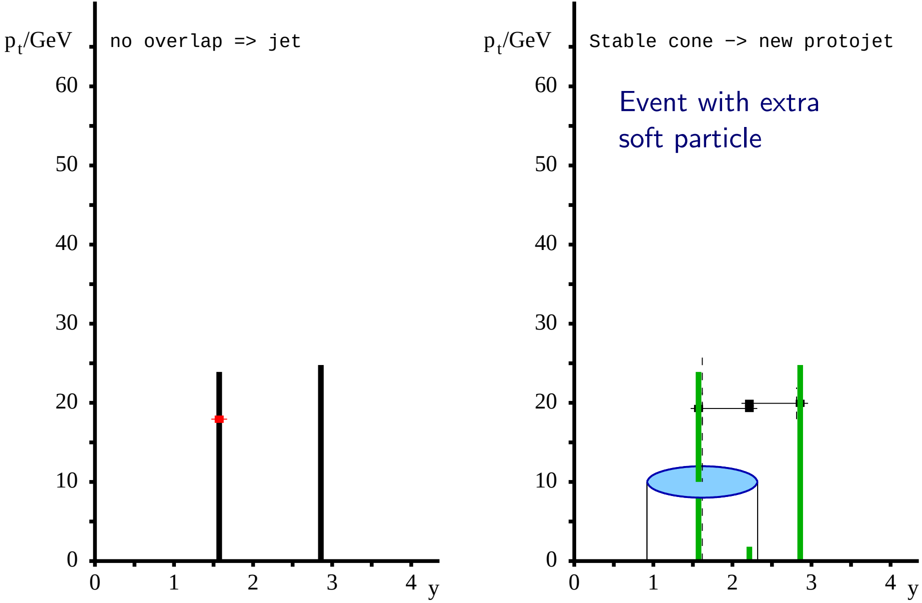




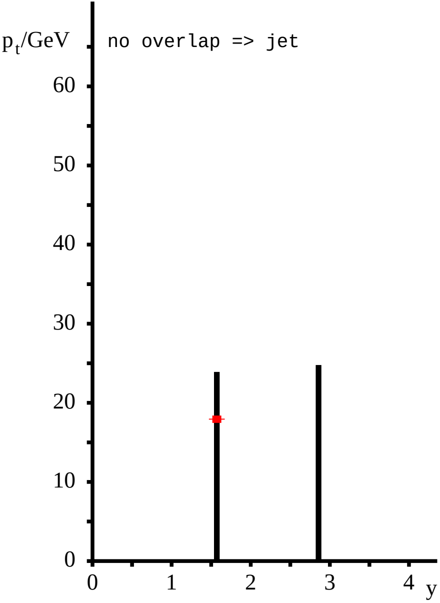
IC-SM: infrared safety

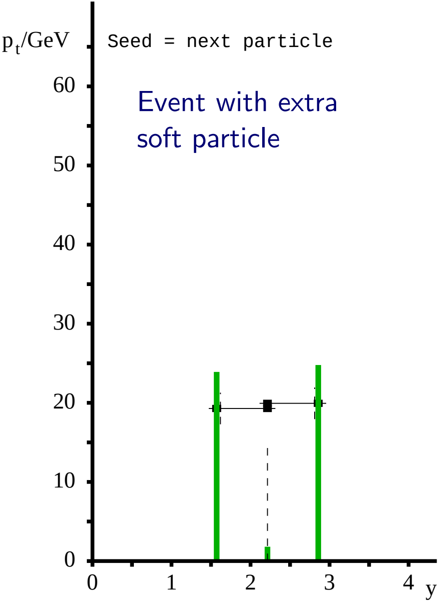


IC-SM: infrared safety

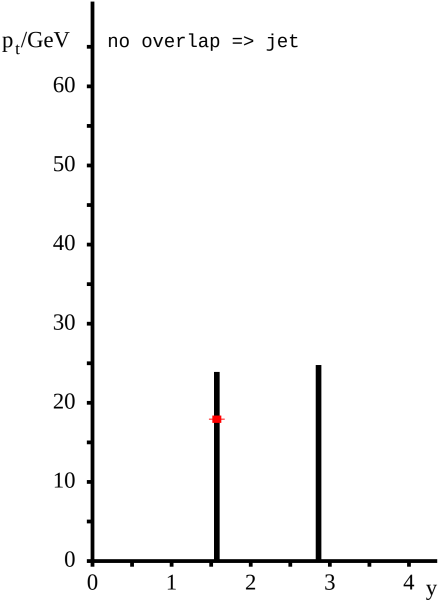


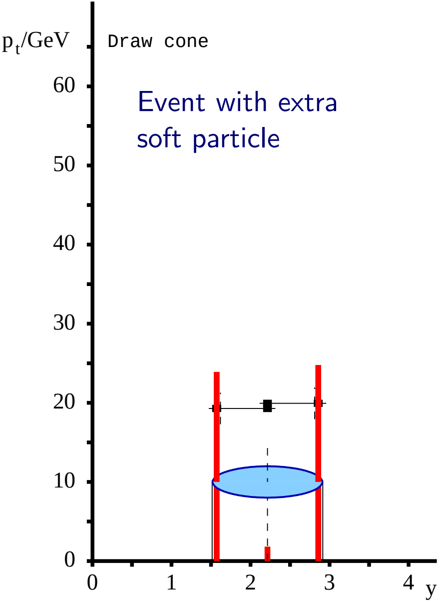
IC-SM: infrared safety



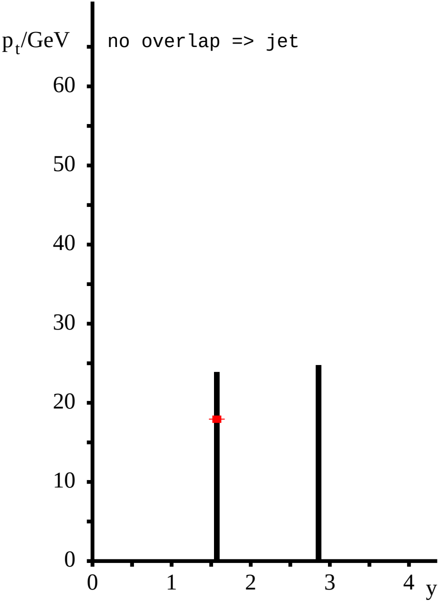


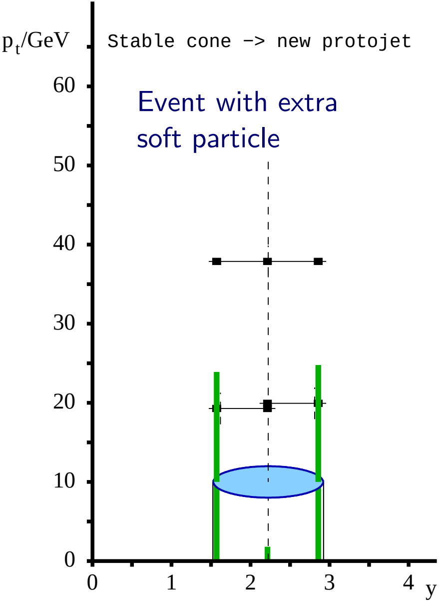
IC-SM: infrared safety



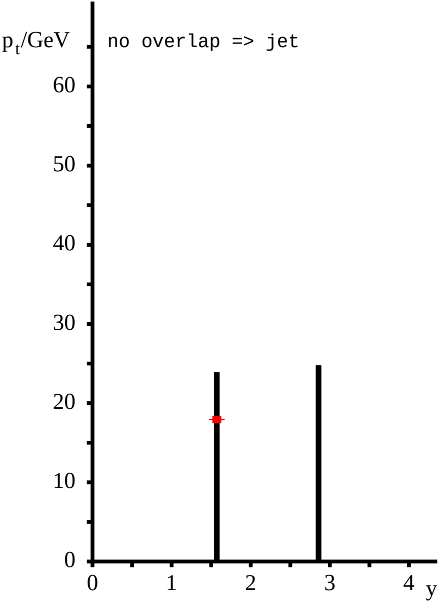


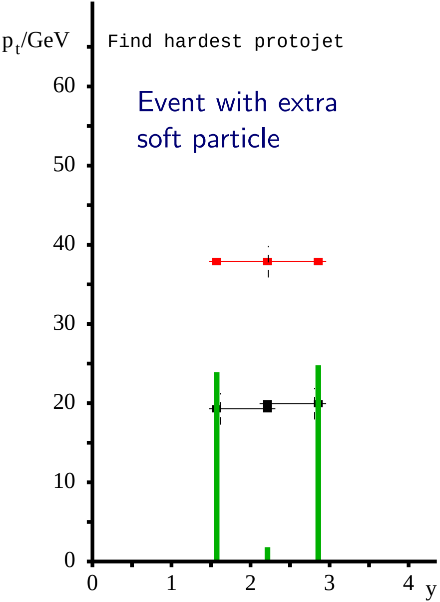
IC-SM: infrared safety



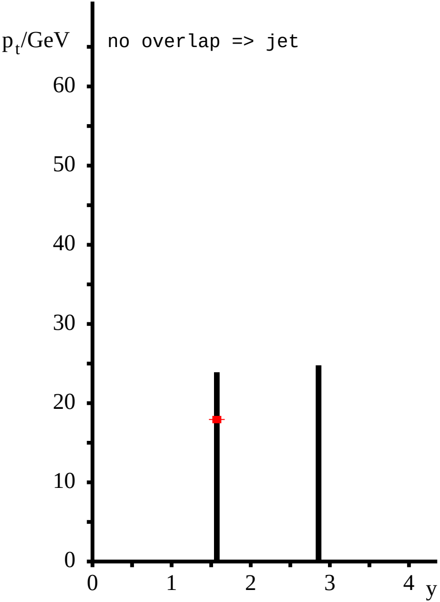


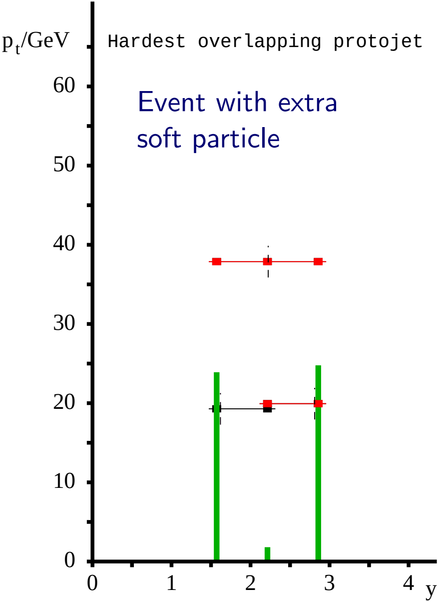
IC-SM: infrared safety



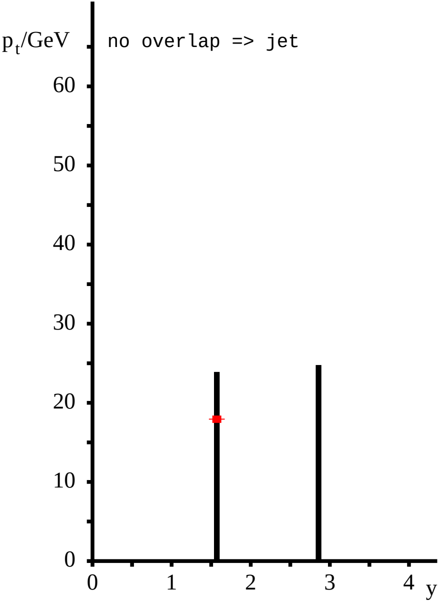


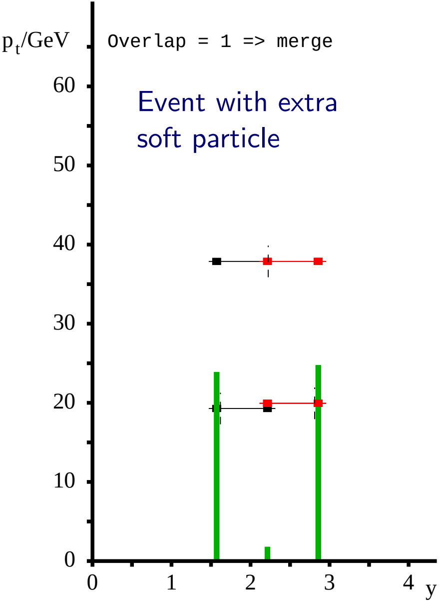
IC-SM: infrared safety



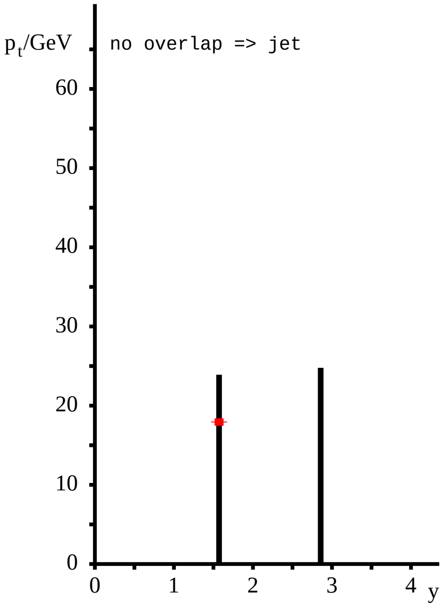


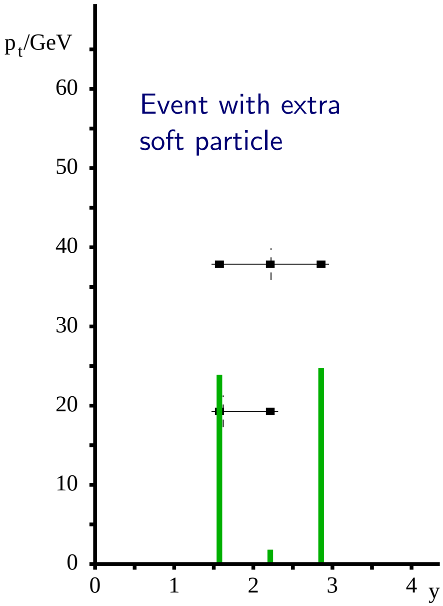
IC-SM: infrared safety



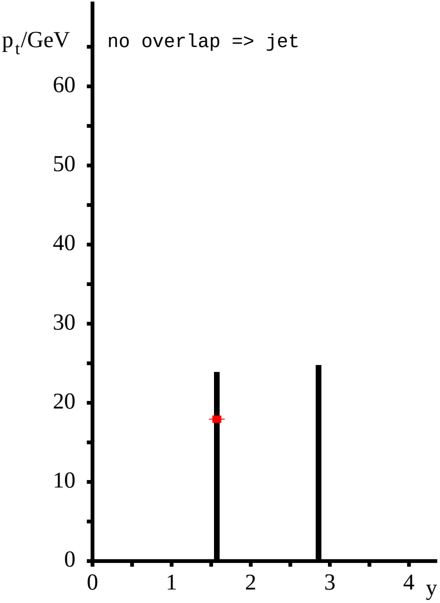


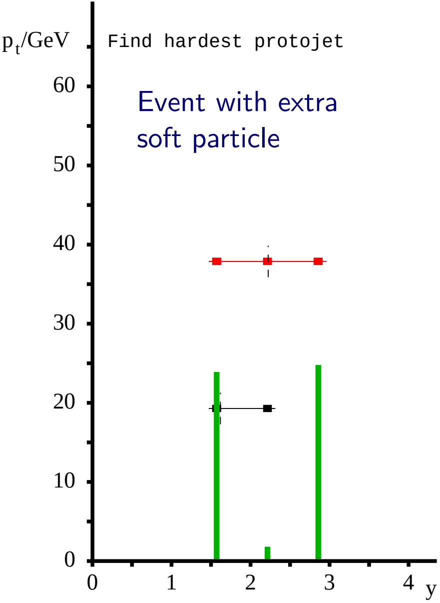
IC-SM: infrared safety



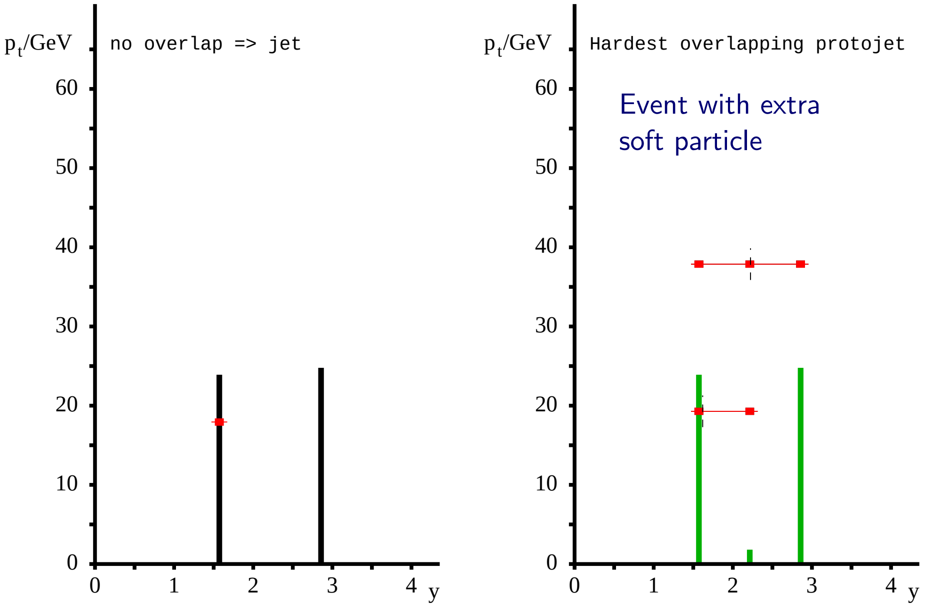


IC-SM: infrared safety

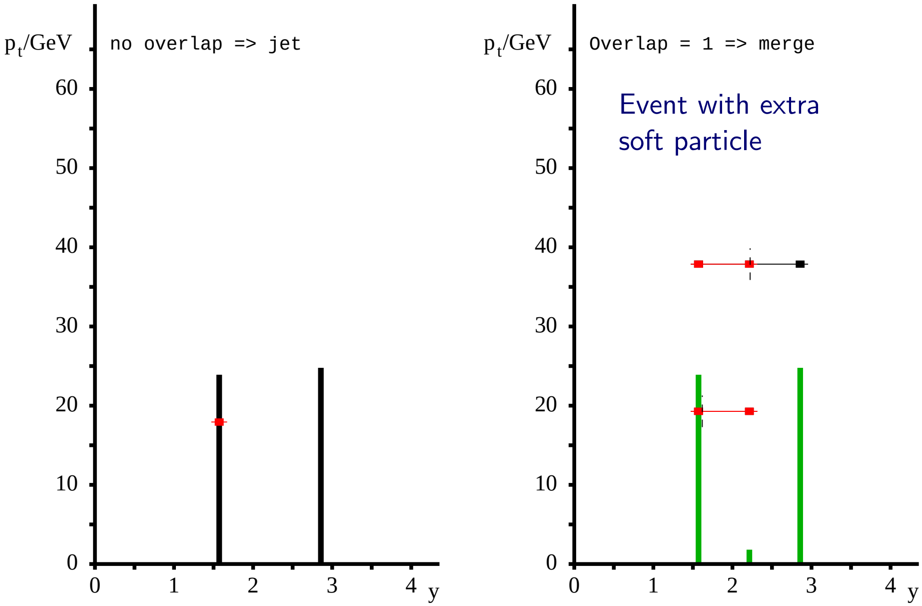




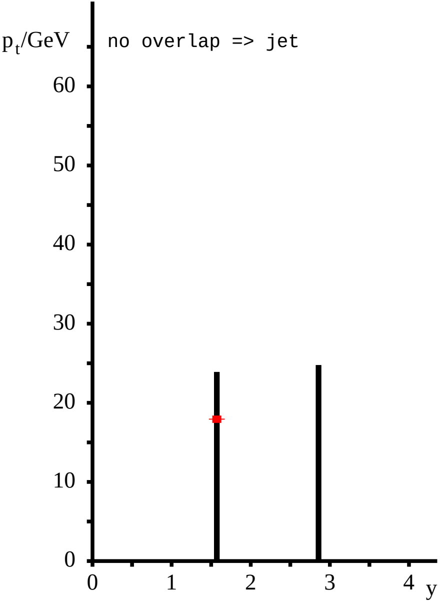
IC-SM: infrared safety

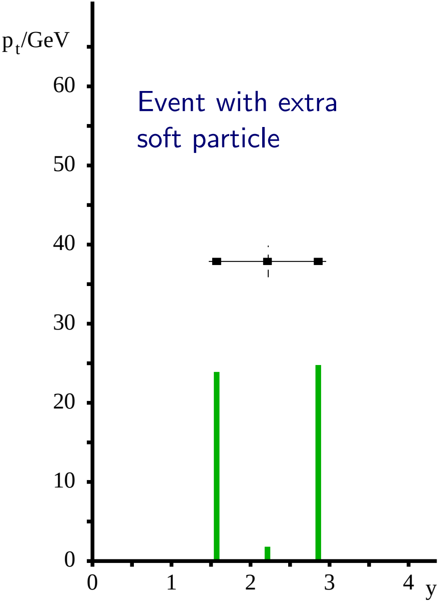


IC-SM: infrared safety

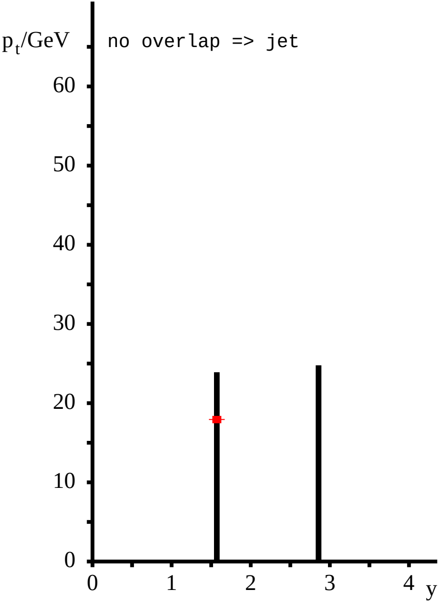


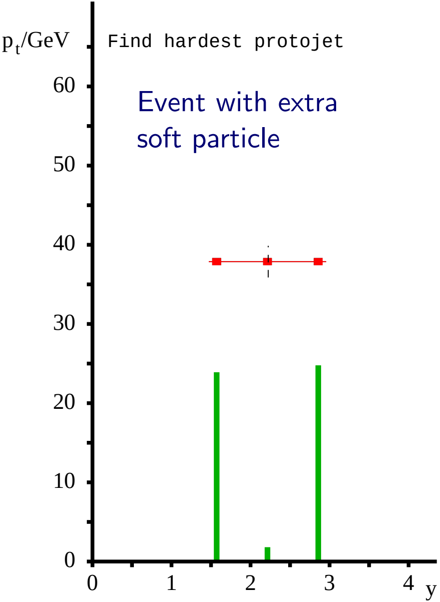
IC-SM: infrared safety





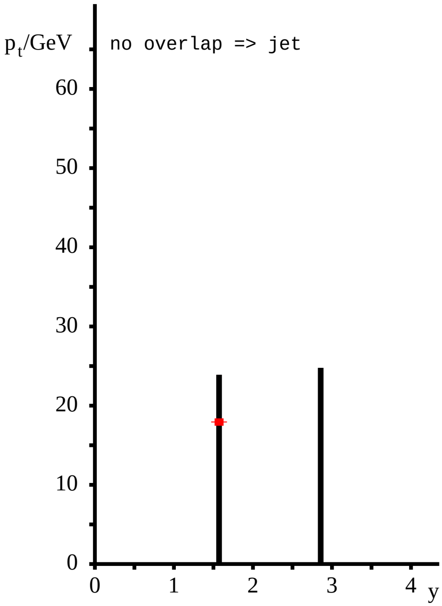
IC-SM: infrared safety

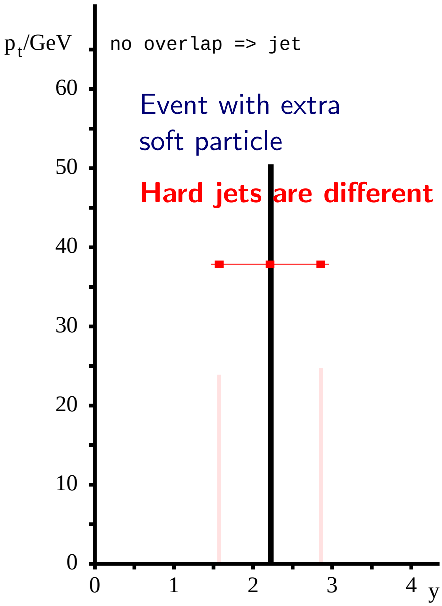




Event with extra soft particle

IC-SM: infrared safety





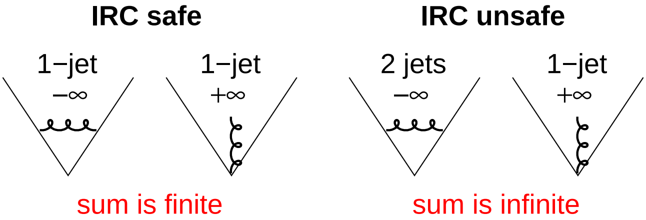
Event with extra soft particle

Hard jets are different

IRC safety crucial for theory

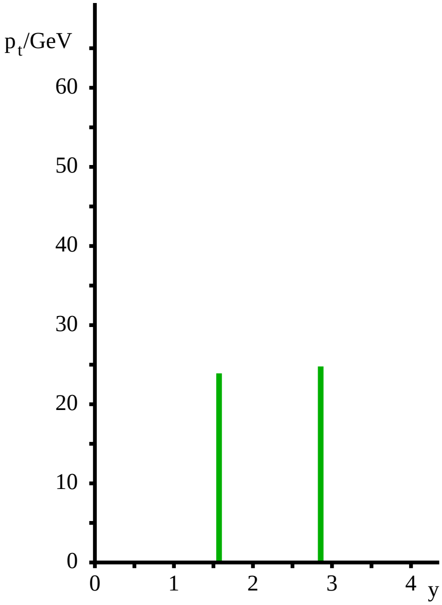
Soft emission, collinear splitting are both **infinite** in pert. QCD.

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



Some calculations simply become **meaningless**

Midpoint algorithm (IC_{mp}-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

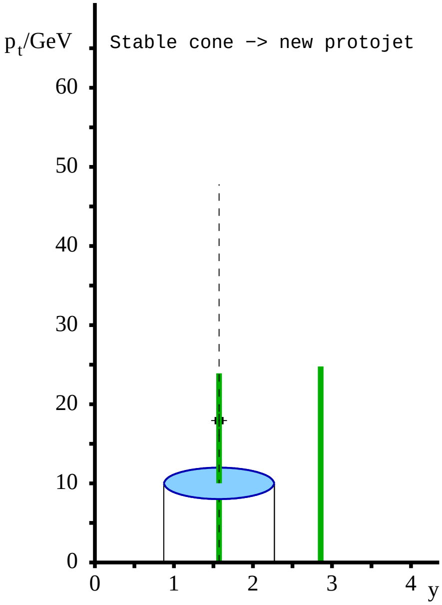
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC$_{mp}$-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

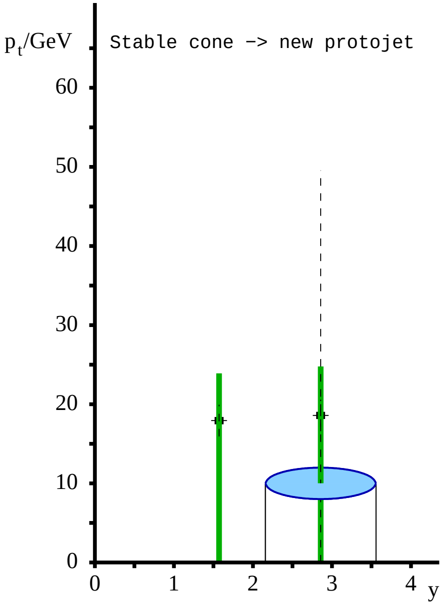
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC$_{mp}$-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

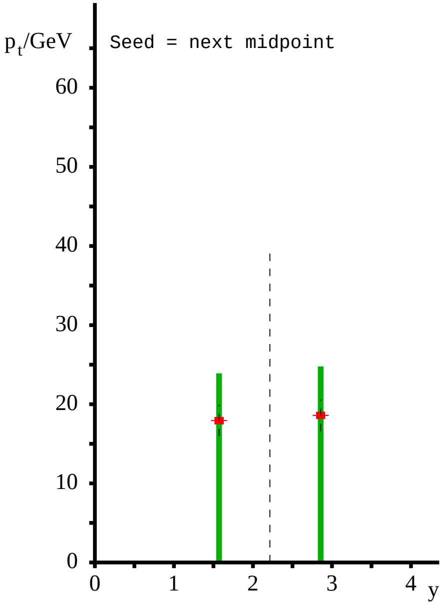
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC_{mp}-SM)



Looking for stable cones ≃ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

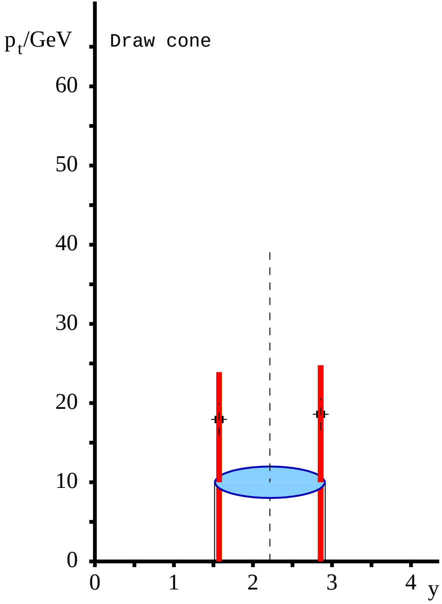
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC$_{mp}$-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

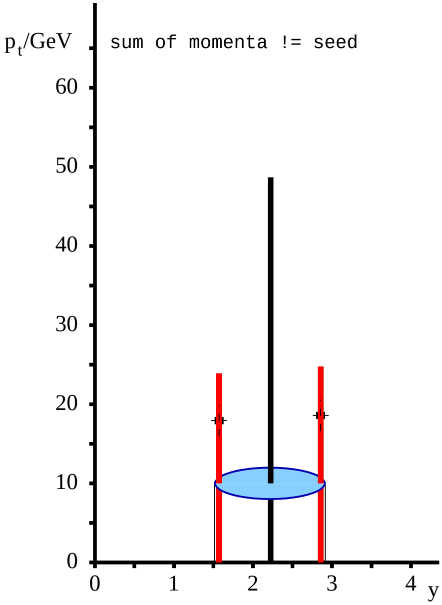
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC_{mp}-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

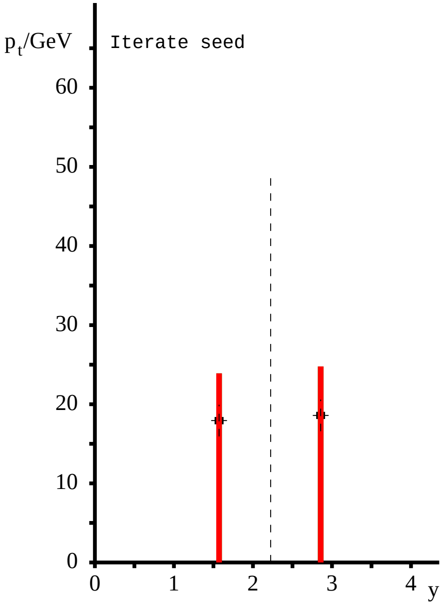
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC_{mp}-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

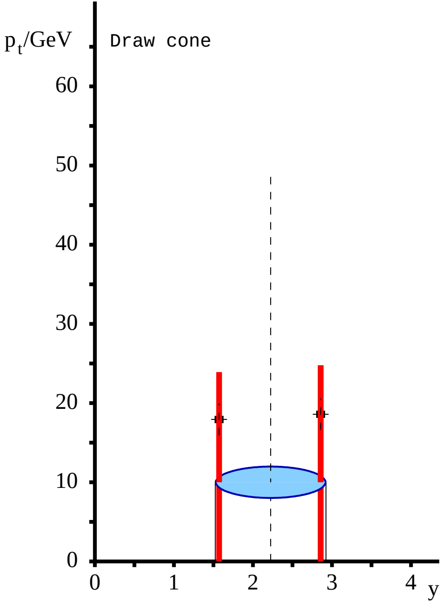
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC$_{mp}$-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

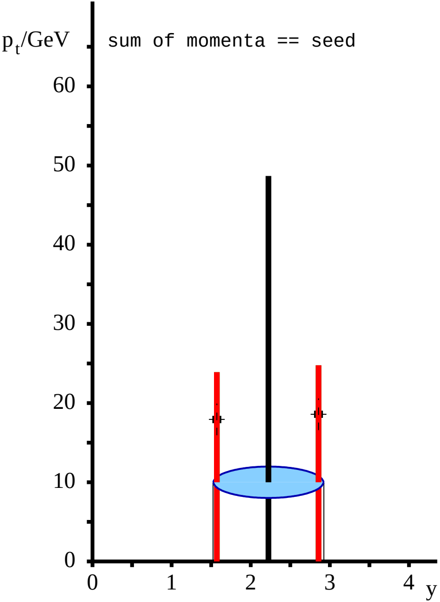
Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

CDF Midpoint algorithm
D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC$_{mp}$-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

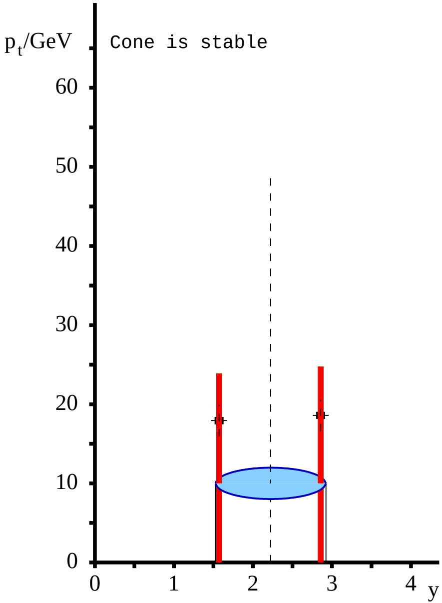
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC$_{mp}$-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

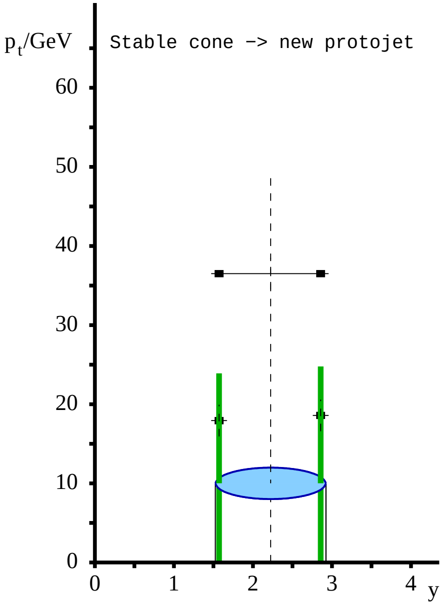
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for
2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC$_{mp}$-SM)



Looking for stable cones ≃ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

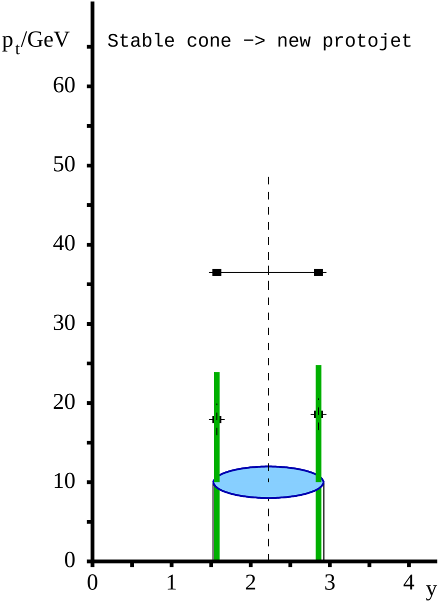
Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

CDF Midpoint algorithm
D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint algorithm (IC_{mp}-SM)



Looking for stable cones $\simeq$ finding local minima of a potential.

Problem: set of iterative solution depends on set of starting points.

Patch: after 1st round of iteration, find midpoints between protojets, use as new seeds

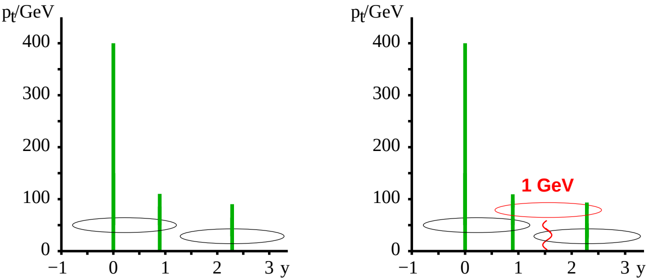
CDF Midpoint algorithm

D0 Run II algorithm

This solves problem for 2-hard-particle configs.

[But it persists for 3-hard]

Midpoint IR problem



Stable cones with midpoint: {1,2} & {3} {1,2} & {2,3} & {3}

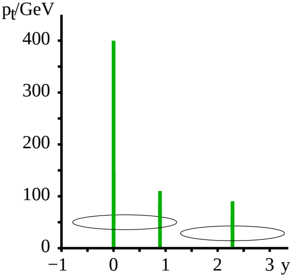
Jets with midpoint ($f = 0.5$) {1,2} & {3} {1,2,3}

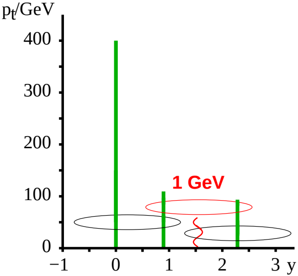
Midpoint cone alg. misses some stable cones; extra soft particle → extra starting point → extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR problem





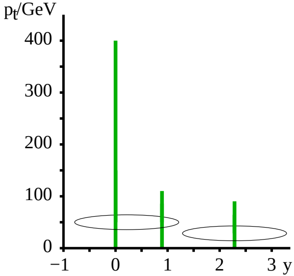
Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

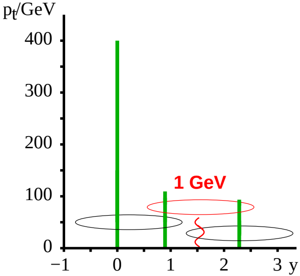
Midpoint cone alg. misses some stable cones; extra soft particle ⟶ extra starting point ⟶ extra stable cone found

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

Does IRC safety really matter?

IRC safety & real-life

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

$$\alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \infty \rightarrow \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \ln p_t/\Lambda \rightarrow \alpha_s^2 + \underbrace{\alpha_s^3 + \alpha_s^3}_{\textbf{BOTH WASTED}}$$

Among consequences of IR unsafety:

	Last meaningful order			
	JetClu, ATLAS cone [IC-SM]	MidPoint [IC$_{mp}$-SM]	CMS it. cone [IC-PR]	Known at
Inclusive jets	LO	NLO	NLO	NLO ($\rightarrow$ 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

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

IRC safety & real-life

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

$$\alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \infty \to \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \ln p_t/\Lambda \to \alpha_s^2 + \underbrace{\alpha_s^3 + \alpha_s^3}_{\text{BOTH WASTED}}$$

Among consequences of IR unsafety:

	Last meaningful order			
	JetClu, ATLAS cone [IC-SM]	MidPoint [IC$_{mp}$-SM]	CMS it. cone [IC-PR]	Known at
Inclusive jets	LO	NLO	NLO	NLO ($\to$ 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]
$m_{\rm jet}$ in $2j + X$	**none**	**none**	**none**	LO

NB: \$30 – 50M 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

IRC safety & real-life

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

$$\alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \infty \to \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \times \ln p_t/\Lambda \to \alpha_s^2 + \underbrace{\alpha_s^3 + \alpha_s^3}_{\text{BOTH WASTED}}$$

Among consequences of IR unsafety:

	Last meaningful order			
	JetClu, ATLAS cone [IC-SM]	MidPoint [IC$_{mp}$-SM]	CMS it. cone [IC-PR]	Known at
Inclusive jets	LO	NLO	NLO	NLO ($\to$ 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

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

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

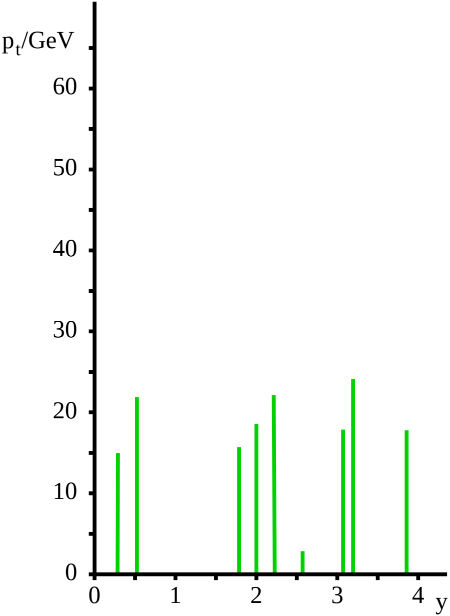
3. High lumi adds lots of extra soft seeds

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

Can we cure this IR safety problem?

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

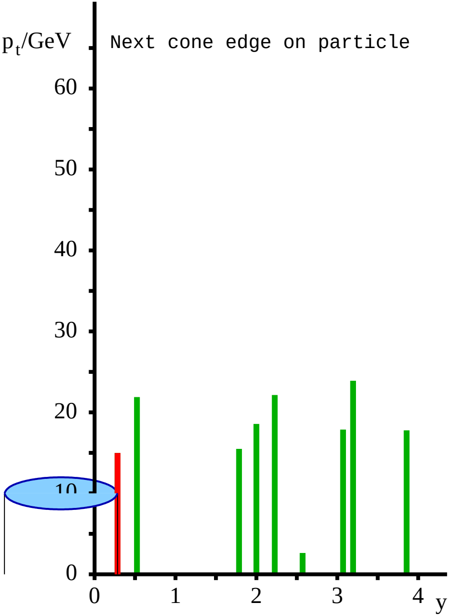
Procedure in 1 dimension (y):

- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**
GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

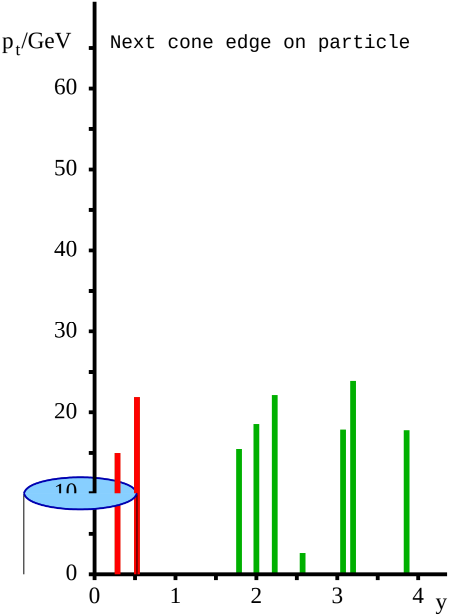
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y, ϕ) *can design analogous procedure* **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

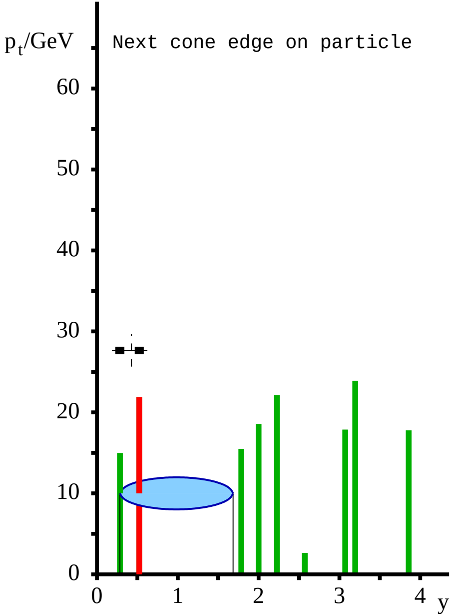
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

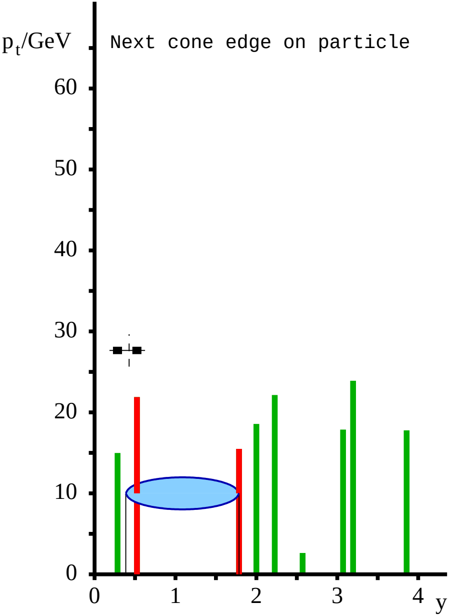
Procedure in 1 dimension (y):

- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y, ϕ) can design analogous procedure **SISCone**
GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

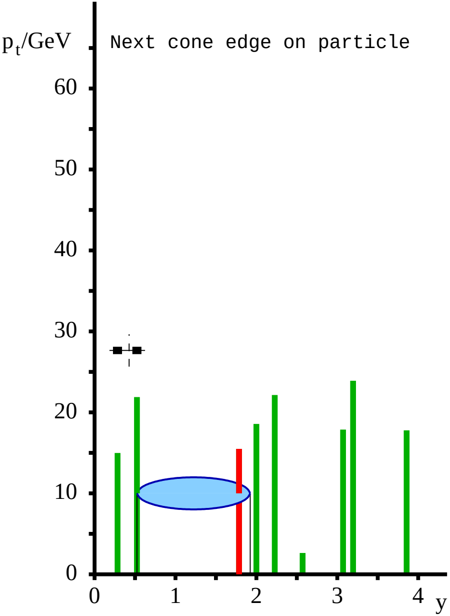
- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

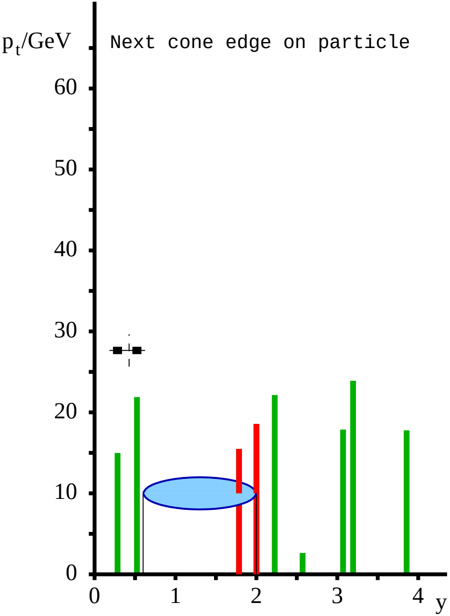
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y, ϕ) *can design analogous procedure* **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

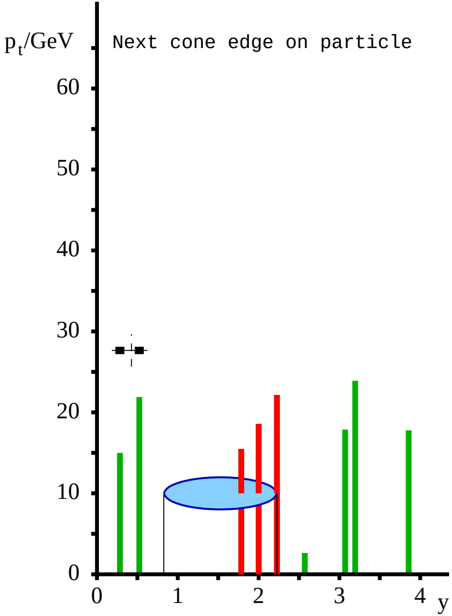
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y, ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

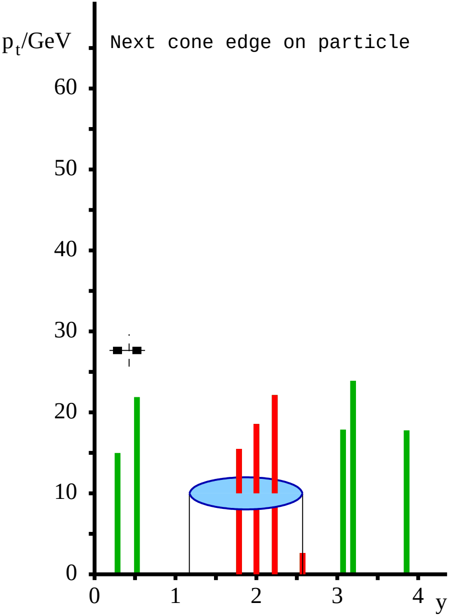
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

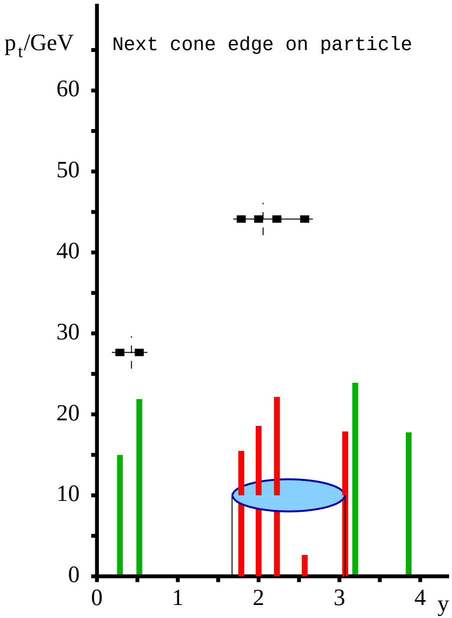
- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

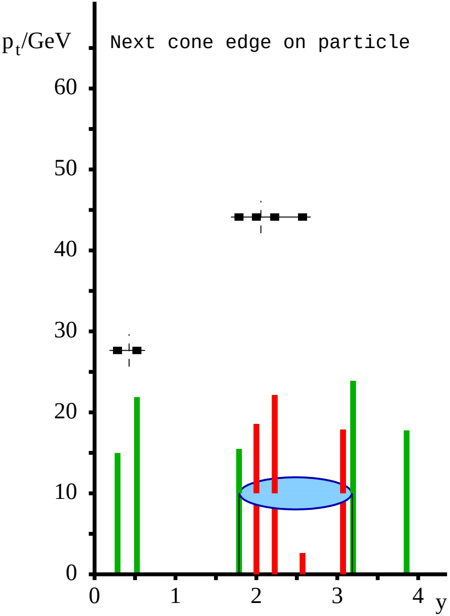
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

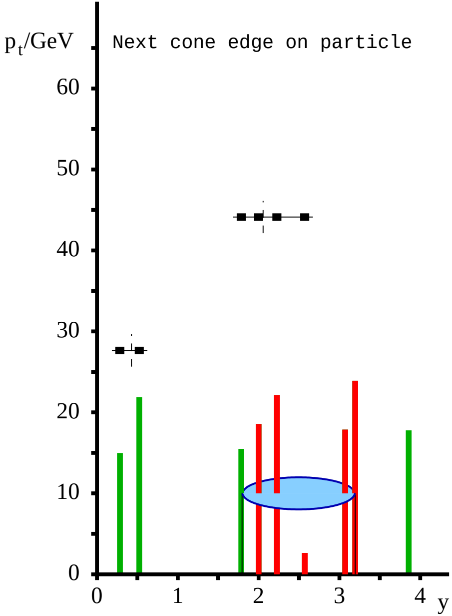
- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y, ϕ) *can design analogous procedure* — **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

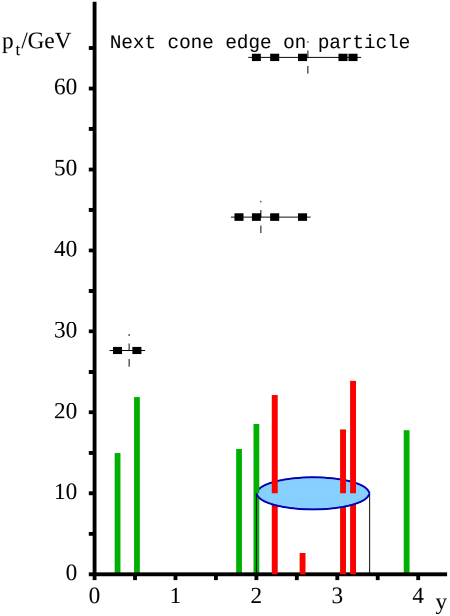
- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

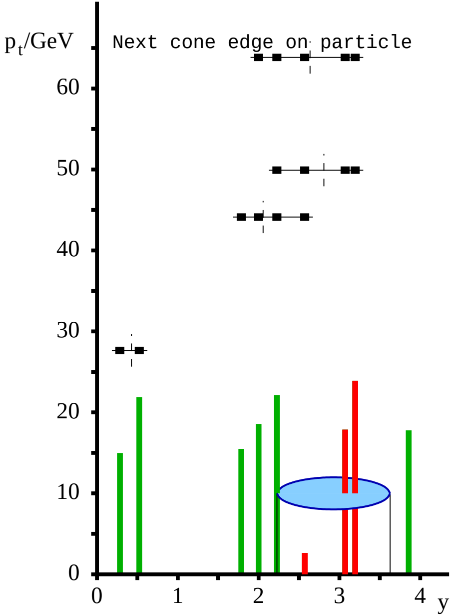
Procedure in 1 dimension (y):

- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure — SISCone
GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

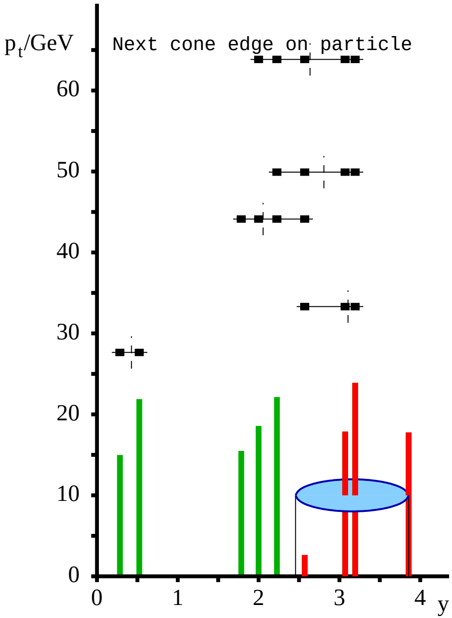
- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y, ϕ) *can design analogous procedure* — SISCone

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

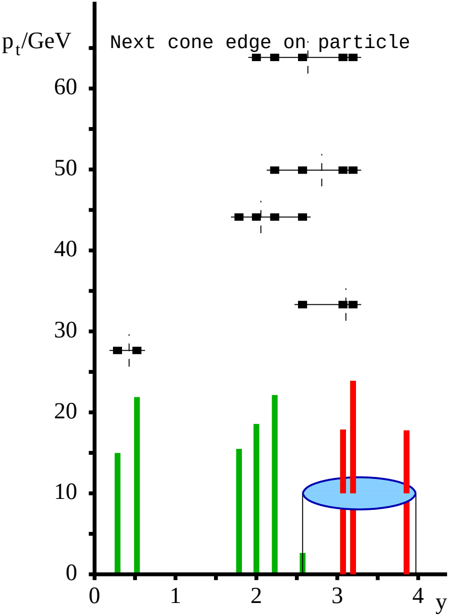
- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

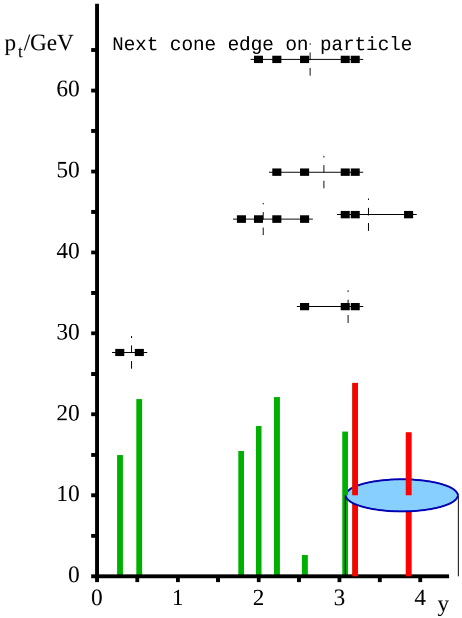
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

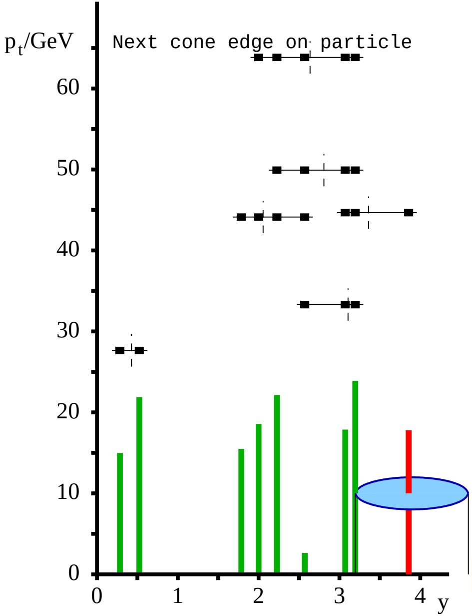
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

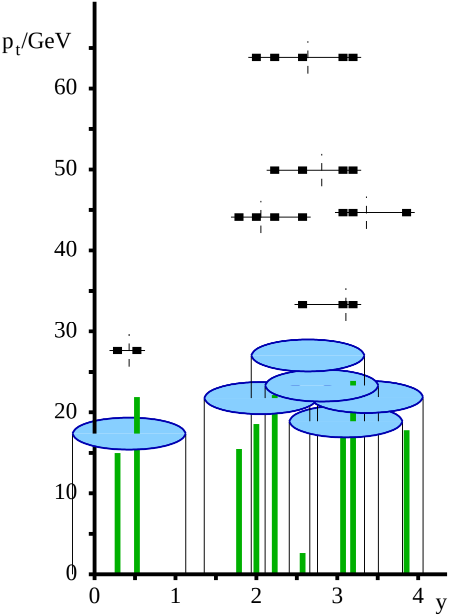
- find all distinct enclosures of radius R by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) *can design analogous procedure* **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

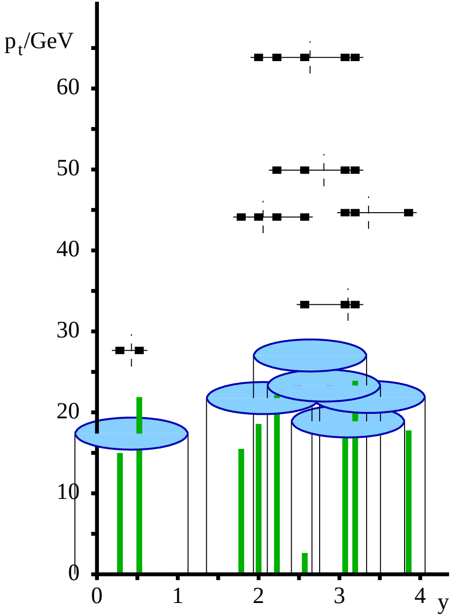
- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

Seedless [Infrared Safe] cones (SC-SM)



Aim to identify *all* stable cones, independently of any seeds

Procedure in 1 dimension (y):

- find all distinct enclosures of radius R — by repeatedly sliding a cone sideways until edge touches a particle
- check each for stability
- then run usual split–merge

In 2 dimensions (y,ϕ) can design analogous procedure **SISCone**

GPS & Soyez '07

This gives an IRC safe cone alg.

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 !

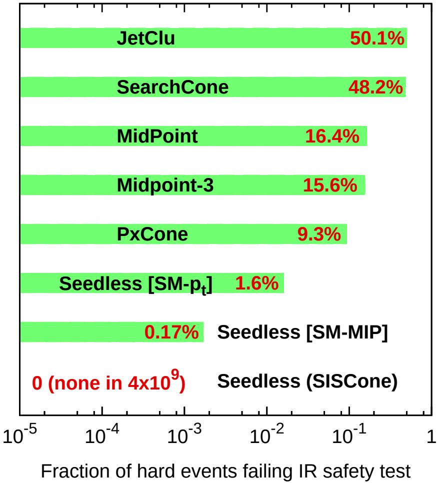
Be careful with split–merge too

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



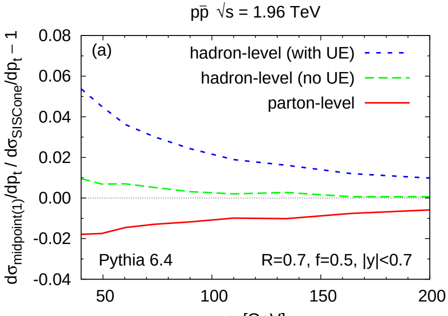
How much does IR safety *really* matter?

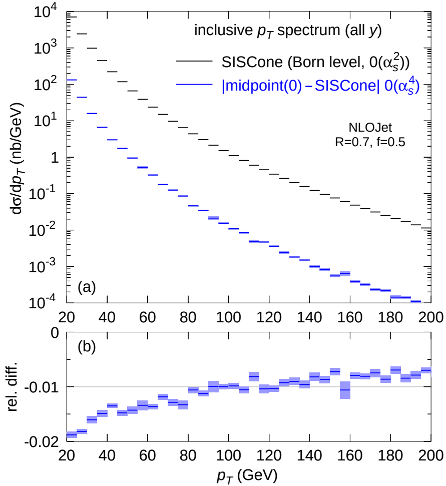
Compare midpoint and SISCone

Result depends on observable:

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

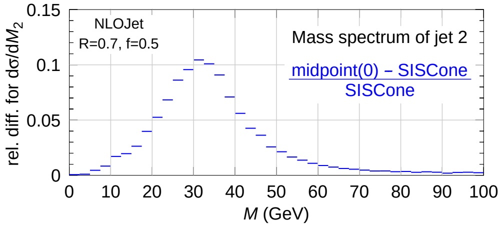
seedless reduces UE effect





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.

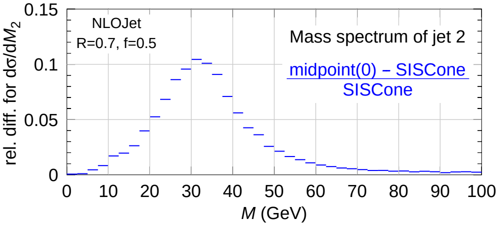
- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$

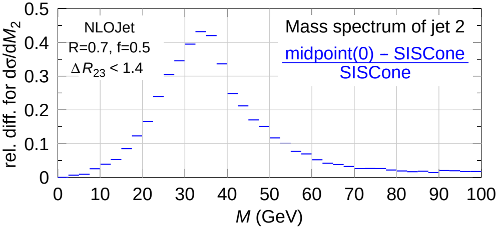
  e.g. for jets from common decay chain

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

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

Bottom line on IR safety

- IR safety often matters less in *inclusive* quantities
- It matters more in multi-jet cases

- ATLAS cone, JetClu (IC-SM) are *very bad*
- CMS cone (IC-PR), Midpoint (IC_{mp}-SM) *moderately bad*

- An IRC safe cone algorithm exists (SISCone)
- **Avoid trouble later: use IR-safe algs from the start**

cf. CDF W+jets

What jet definition should I use?

[jet def. $\equiv$ jet alg., R, (f)]

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 ordering	$N \ln N$
$p = -1$	anti-k_t Cacciari, GPS, Soyez '08 $\sim$ reverse-k_t Delsart	Hierarchy meaningless, jets like CMS cone (IC-PR)	$N^{3/2}$
SC-SM	SISCone GPS Soyez '07 + Tevatron run II '00	Replaces JetClu, ATLAS MidPoint (xC-SM) cones	$N^2 \ln N$ exp.

Compromise between having a limited set of algs.
and a good range of complementary properties

COMMERCIAL BREAK

Use FastJet — it's free!

One place to stop for all your jet-finding needs:

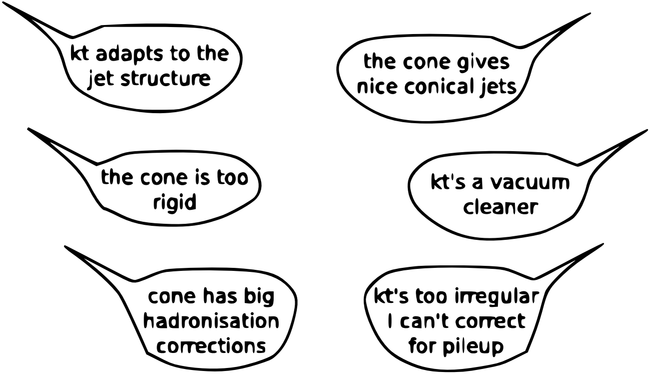
FASTJET

http://www.lpthe.jussieu.fr/~salam/fastjet
Cacciari, GPS & Soyez '05–07

- Fast, native, computational-geometry methods for k_t, Cam/Aachen
  Cacciari & GPS '05-06
- Plugins for SISCone (plus some other, deprecated cones)
- Many other features too, e.g. jet areas

Jet Folklore

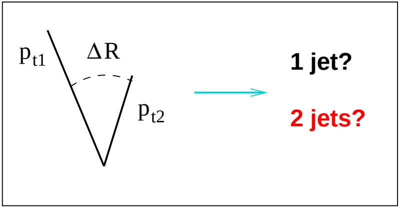
Jet discussions: often polarised, driven by unquantified statements



- Rigorous approach is to quantify similarities & differences
- Bottom line: grains of truth in the qualitative statements

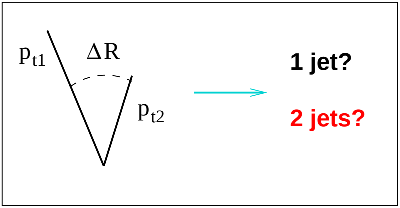
  So want good cone algorithms too [NB: recall, two variants xC-SM & xC-PR]

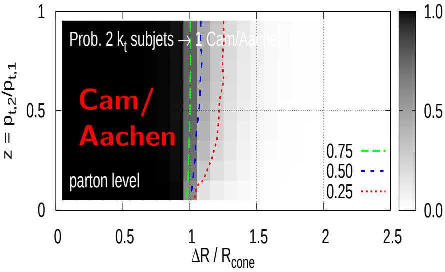
the *reach* of jet algorithms



SISCone (xC-SM) reaches further for hard radiation than other algs

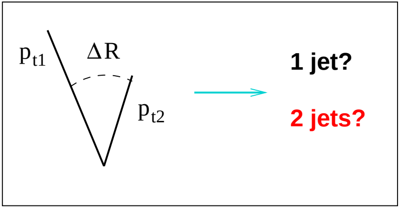
the *reach* of jet algorithms

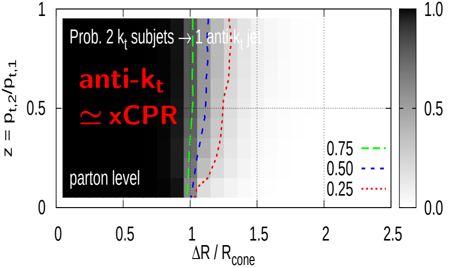


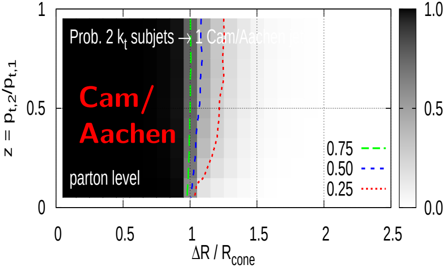


SISCone (xC-SM) reaches further for hard radiation than other algs

the *reach* of jet algorithms

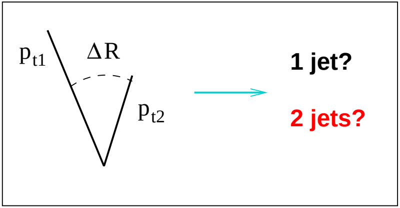


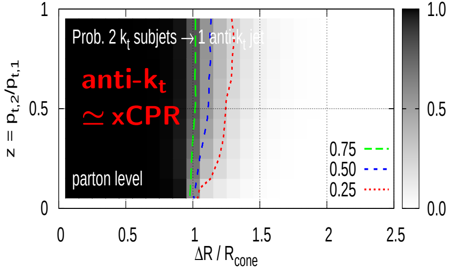


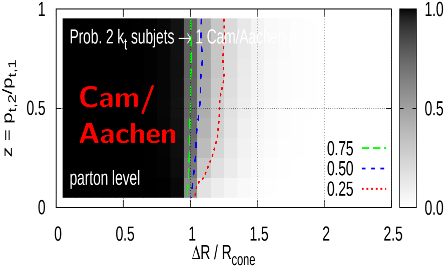


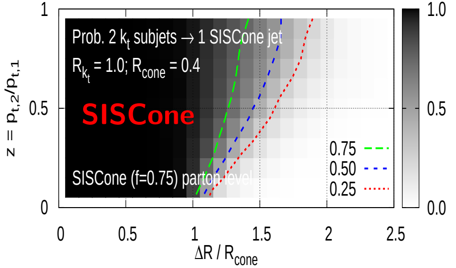
SISCone (xC-SM) reaches further for hard radiation than other algs

the *reach* of jet algorithms



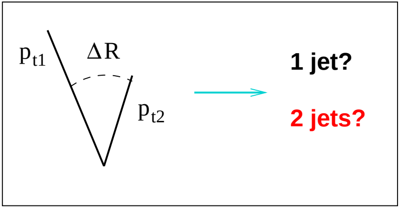


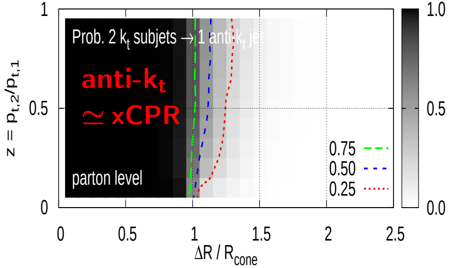


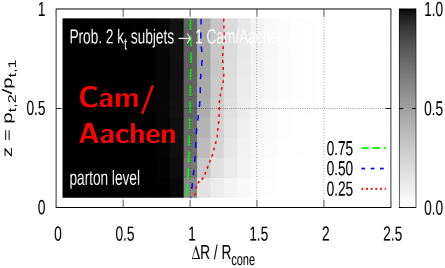


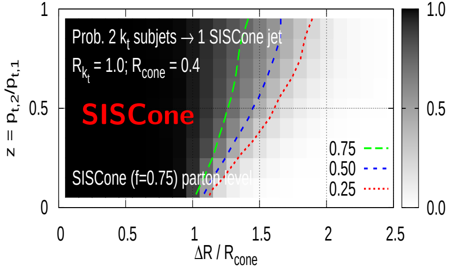
SISCone (xC-SM) reaches further for hard radiation than other algs

the *reach* of jet algorithms



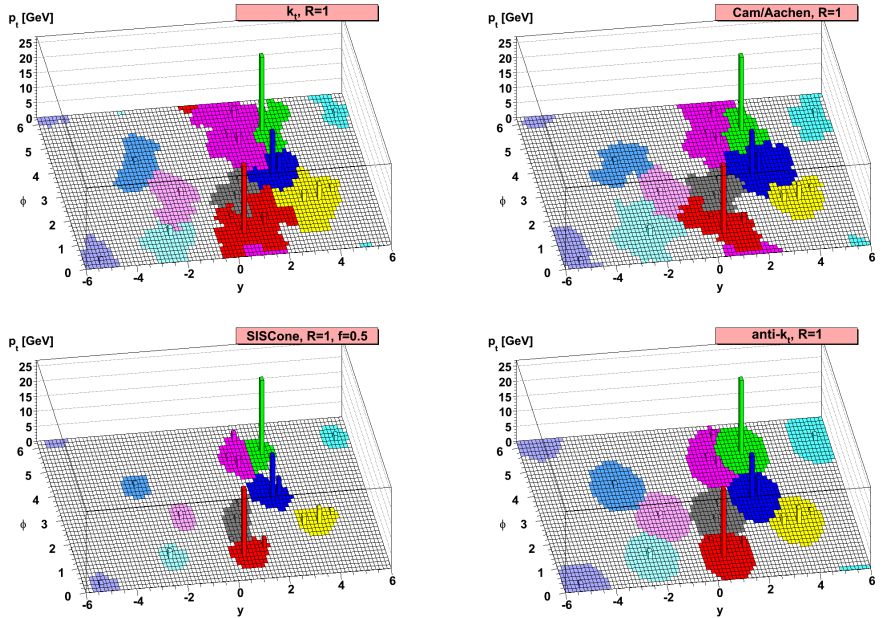






SISCone (xC-SM) reaches further for hard radiation than other algs

Jet contours – visualised



To first approx:
various algs. moderately different;
but **R** can matter a lot more

4-way tension in many measurements:

Prefer small R	prefer large R
resolve many jets (e.g. $t\bar{t}$)	minimize QCD radiation loss
limit UE & pileup	limit hadronisation

Jets v. R

Parton $p_t \to$ jet p_t

Ill-defined: MC "parton"

PT radiation:

$$q: \quad \Delta p_t \simeq \frac{\alpha_s C_F}{\pi} p_t \ln R$$

$$g: \quad \Delta p_t \simeq \frac{\alpha_s C_A}{\pi} p_t \ln R$$

Hadronisation:

$$q: \quad \Delta p_t \simeq \frac{C_F}{R} \cdot 0.4 \text{ GeV}$$

$$g: \quad \Delta p_t \simeq \frac{C_A}{R} \cdot 0.4 \text{ GeV}$$

Underlying event:

$$q, g: \quad \Delta p_t \simeq \frac{R^2}{2} \cdot 2.5\text{–}15 \text{ GeV}$$

crude analytical estimates
cf. Dasgupta, Magnea & GPS '07

Jets v. R

Parton $p_t \longrightarrow$ jet p_t

Ill-defined: MC "parton"

PT radiation:

$$q: \quad \Delta p_t \simeq \frac{\alpha_s C_F}{\pi} p_t \ln R$$

$$g: \quad \Delta p_t \simeq \frac{\alpha_s C_A}{\pi} p_t \ln R$$

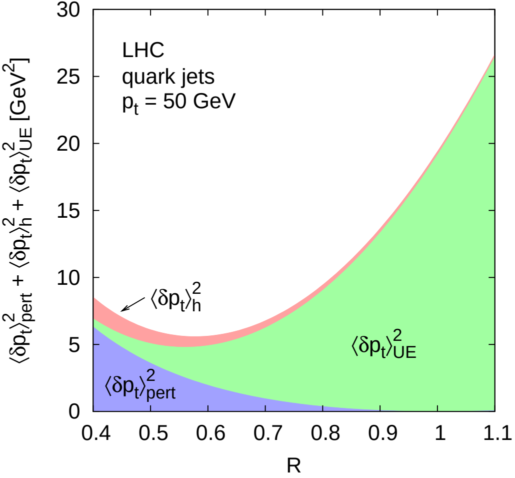
Hadronisation:

$$q: \quad \Delta p_t \simeq \frac{C_F}{R} \cdot 0.4 \text{ GeV}$$

$$g: \quad \Delta p_t \simeq \frac{C_A}{R} \cdot 0.4 \text{ GeV}$$

Underlying event:

$$q,g: \quad \Delta p_t \simeq \frac{R^2}{2} \cdot 2.5\text{—}15 \text{ GeV}$$



crude analytical estimates

cf. Dasgupta, Magnea & GPS '07

Jets v. R

Parton $p_t \longrightarrow$ jet p_t

Ill-defined: MC "parton"

PT radiation:

$$q: \quad \Delta p_t \simeq \frac{\alpha_s C_F}{\pi} p_t \ln R$$

$$g: \quad \Delta p_t \simeq \frac{\alpha_s C_A}{\pi} p_t \ln R$$

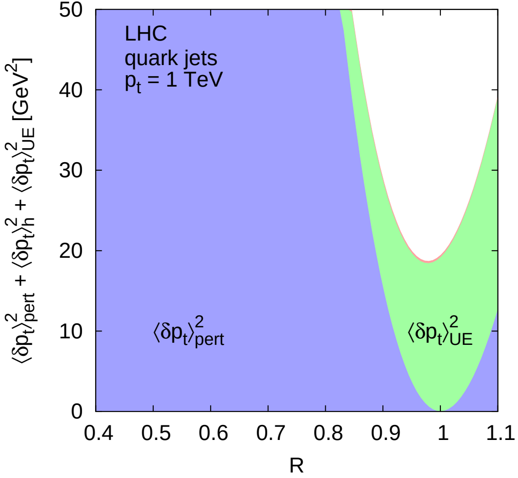
Hadronisation:

$$q: \quad \Delta p_t \simeq \frac{C_F}{R} \cdot 0.4 \text{ GeV}$$

$$g: \quad \Delta p_t \simeq \frac{C_A}{R} \cdot 0.4 \text{ GeV}$$

Underlying event:

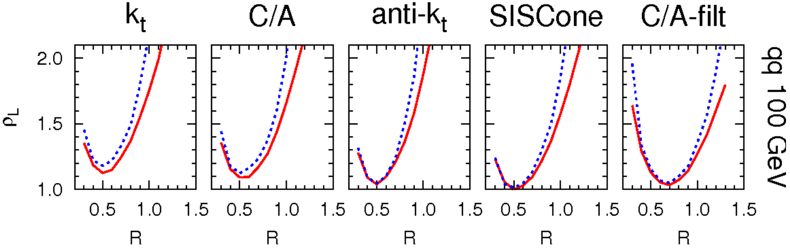
$$q, g: \quad \Delta p_t \simeq \frac{R^2}{2} \cdot 2.5 - 15 \text{ GeV}$$



crude analytical estimates

cf. Dasgupta, Magnea & GPS '07

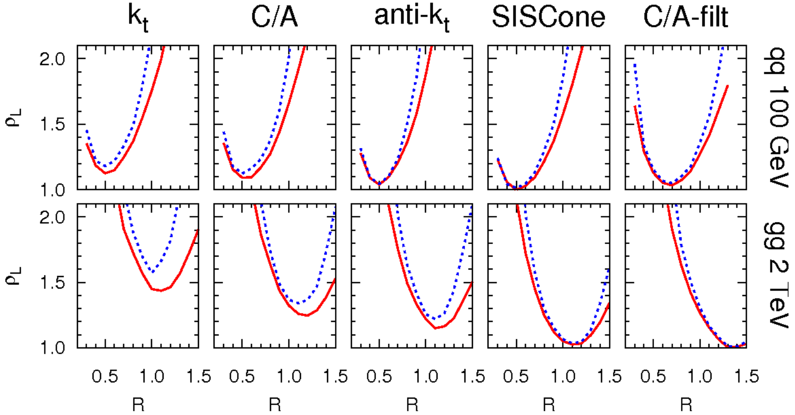
Relative peak quality (lumi ratios ρ_L), LHC



PRELIMINARY

Cacciari, Rojo, GPS & Soyez '08

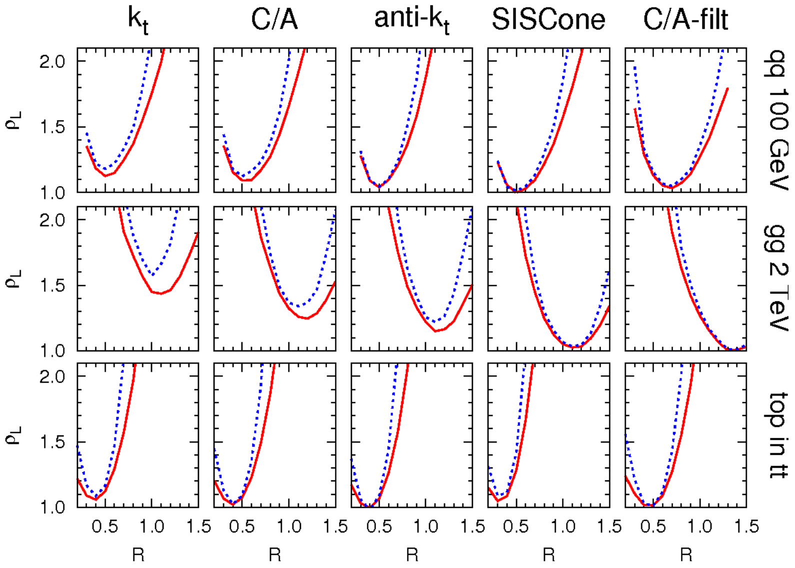
Relative peak quality (lumi ratios ρ_L), LHC



PRELIMINARY

Cacciari, Rojo, GPS & Soyez '08

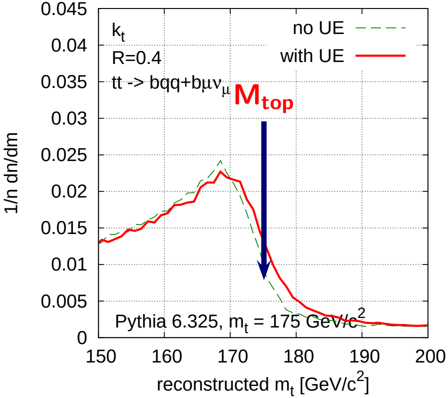
Relative peak quality (lumi ratios ρ_L), LHC



PRELIMINARY

Cacciari, Rojo, GPS & Soyez '08

Robustness: M_{top} varies with R?



<u>Game: measure top mass to 1 GeV</u>

example for Tevatron

$m_t = 175$ GeV

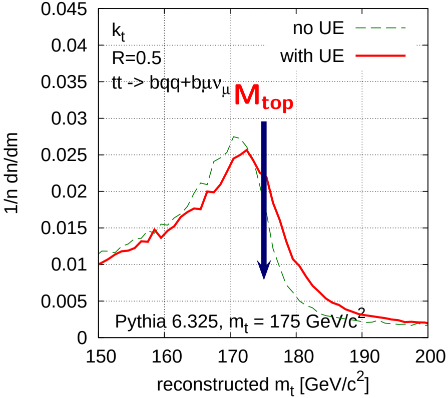
- Small R: lose 6 GeV to PT radiation and hadronisation, UE and pileup irrelevant
- Large R: hadronisation and PT radiation leave mass at ~ 175 GeV, UE adds $2-4$ GeV.

Is the final top mass (after W, jet-energy-scale and Monte Carlo unfolding) independent of R used to measure jets?

Powerful cross-check of systematic effects

cf. Seymour & Tevlin '06

Robustness: M_{top} varies with R?



Game: measure top mass to 1 GeV

example for Tevatron

$m_t = 175$ GeV

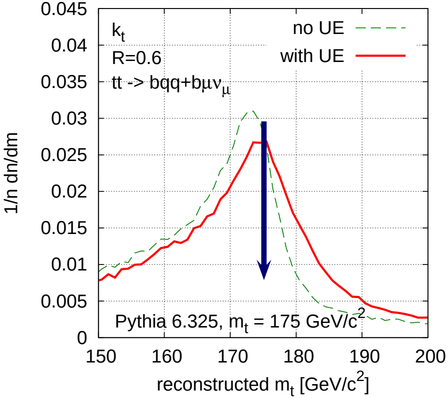
- Small R: lose 6 GeV to PT radiation and hadronisation, UE and pileup irrelevant
- Large R: hadronisation and PT radiation leave mass at ~ 175 GeV, UE adds $2 - 4$ GeV.

Is the final top mass (after W, jet-energy-scale and Monte Carlo unfolding) independent of R used to measure jets?

Powerful cross-check of systematic effects

cf. Seymour & Tevlin '06

Robustness: M_{top} varies with R?



Game: measure top mass to 1 GeV

example for Tevatron

$m_t = 175$ GeV

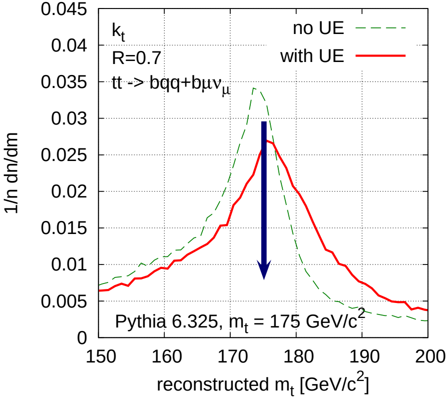
- Small R: lose 6 GeV to PT radiation and hadronisation, UE and pileup irrelevant
- Large R: hadronisation and PT radiation leave mass at ~ 175 GeV, UE adds $2 - 4$ GeV.

Is the final top mass (after W jet-energy-scale and Monte Carlo unfolding) independent of R used to measure jets?

Powerful cross-check of systematic effects

cf. Seymour & Tevlin '06

Robustness: M_{top} varies with R?



Game: measure top mass to 1 GeV

example for Tevatron

$m_t = 175$ GeV

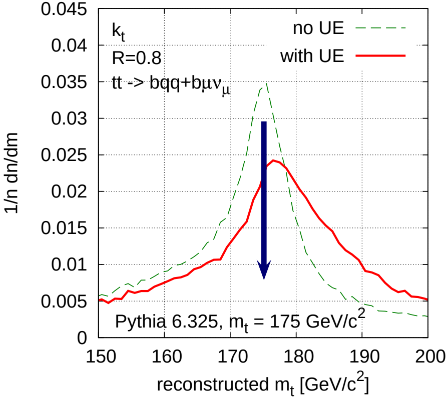
- Small R: lose 6 GeV to PT radiation and hadronisation, UE and pileup irrelevant
- Large R: hadronisation and PT radiation leave mass at ~ 175 GeV, UE adds $2-4$ GeV.

Is the final top mass (after W jet-energy-scale and Monte Carlo unfolding) independent of R used to measure jets?

Powerful cross-check of systematic effects

cf. Seymour & Tevlin '06

Robustness: M_{top} varies with R?



<u>Game: measure top mass to 1 GeV</u>

example for Tevatron

$m_t = 175$ GeV

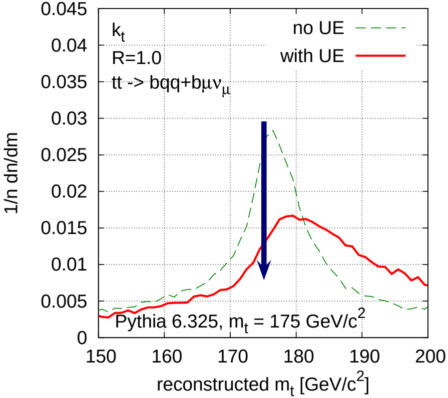
- Small R: lose 6 GeV to PT radiation and hadronisation, UE and pileup irrelevant
- Large R: hadronisation and PT radiation leave mass at ~ 175 GeV, UE adds $2-4$ GeV.

Is the final top mass (after W jet-energy-scale and Monte Carlo unfolding) independent of R used to measure jets?

Powerful cross-check of systematic effects

cf. Seymour & Tevlin '06

Robustness: M_{top} varies with R?



Game: measure top mass to 1 GeV

example for Tevatron

$m_t = 175$ GeV

- Small R: lose 6 GeV to PT radiation and hadronisation, UE and pileup irrelevant
- Large R: hadronisation and PT radiation leave mass at ~ 175 GeV, UE adds $2-4$ GeV.

Is the final top mass (after W jet-energy-scale and Monte Carlo unfolding) independent of R used to measure jets?

Powerful cross-check of systematic effects

cf. Seymour & Tevlin '06

Jets without hard partons:

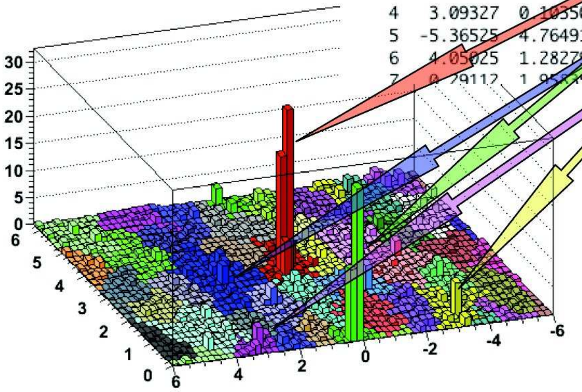
Most jet algorithms give you $\sim 50 - 100$ "jets," mostly not hard.

provide window on UE and min-bias

Making use of *all* jets

```
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

Pushing jets to their limit:

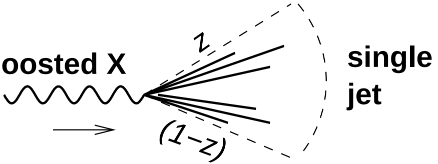
when a W, Z, H or a top $\longrightarrow$ a single jet

Not unusual at LHC: $m_W, m_t \ll 14$ TeV

EW bosons at @ high p_t

Illustrate LHC challenges with a recently widely discussed class of problems:

Can you identify hadronically decaying EW bosons when they're produced at high p_t?



$$R \gtrsim \frac{m}{p_t} \frac{1}{\sqrt{z(1-z)}}$$

Significant discussion over years: **heavy new things decay to EW states**

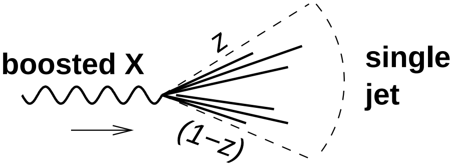
- Seymour '94 [Higgs $\to WW \to \nu\ell$jets]
- Butterworth, Cox & Forshaw '02 [$WW \to WW \to \nu\ell$jets]
- Agashe et al. '06 [KK excitation of gluon $\to t\bar{t}$]
- Butterworth, Ellis & Raklev '07 [SUSY decay chains $\to W, H$]
- Skiba & Tucker-Smith '07 [vector quarks]
- Lilli, Randall & Wang '07 [KK excitation of gluon $\to t\bar{t}$]

ETC.

EW bosons at @ high p_t

Illustrate LHC challenges with a recently widely discussed class of problems:

Can you identify hadronically decaying EW bosons when they're produced at high $\mathbf{p_t}$?



$$R \gtrsim \frac{m}{p_t} \frac{1}{\sqrt{z(1-z)}}$$

Significant discussion over years: **heavy new things decay to EW states**

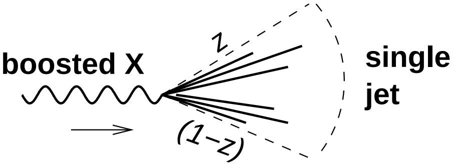
- Seymour '94 [Higgs $\to WW \to \nu\ell$jets]
- Butterworth, Cox & Forshaw '02 [$WW \to WW \to \nu\ell$jets]
- Agashe et al. '06 [KK excitation of gluon $\to t\bar{t}$]
- Butterworth, Ellis & Raklev '07 [SUSY decay chains $\to W, H$]
- Skiba & Tucker-Smith '07 [vector quarks]
- Lilli, Randall & Wang '07 [KK excitation of gluon $\to t\bar{t}$]

ETC.

EW bosons at @ high p_t

Illustrate LHC challenges with a recently widely discussed class of problems:

Can you identify hadronically decaying EW bosons when they're produced at high p_t?



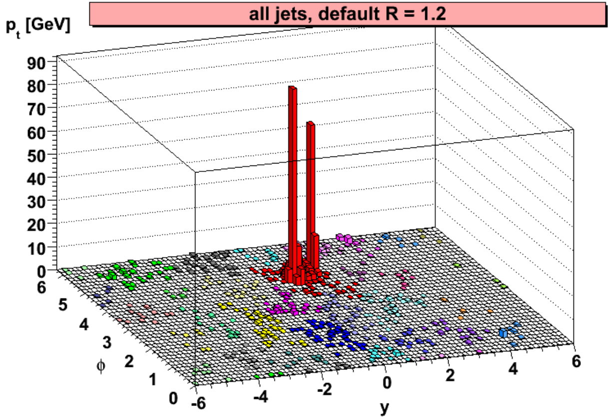
$$R \gtrsim \frac{m}{p_t} \frac{1}{\sqrt{z(1-z)}}$$

Significant discussion over years: **heavy new things decay to EW states**

- Seymour '94 [Higgs $\to WW \to \nu\ell$jets]
- Butterworth, Cox & Forshaw '02 [$WW \to WW \to \nu\ell$jets]
- Agashe et al. '06 [KK excitation of gluon $\to t\bar{t}$]
- Butterworth, Ellis & Raklev '07 [SUSY decay chains $\to W, H$]
- Skiba & Tucker-Smith '07 [vector quarks]
- Lilli, Randall & Wang '07 [KK excitation of gluon $\to t\bar{t}$]

ETC.

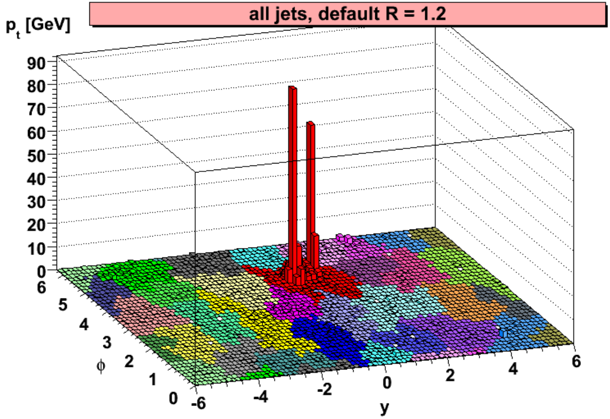
$pp \longrightarrow ZH \longrightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV



[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]
Butterworth, Davison, Rubin & GPS '08

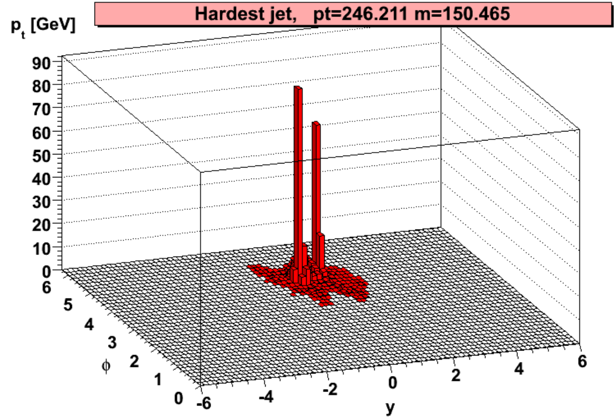
Possible new (light) Higgs discovery channel

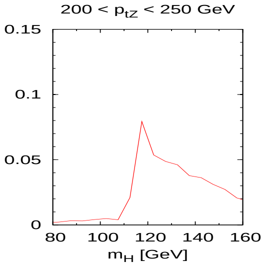
$pp \longrightarrow ZH \longrightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV

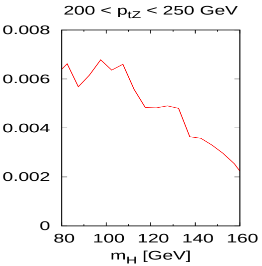


[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]
Butterworth, Davison, Rubin & GPS '08

$pp \longrightarrow ZH \longrightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV



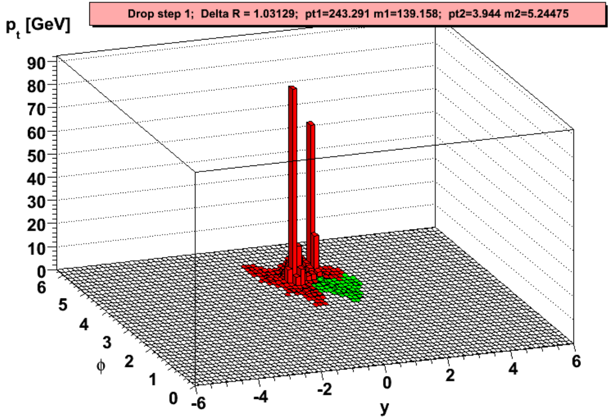




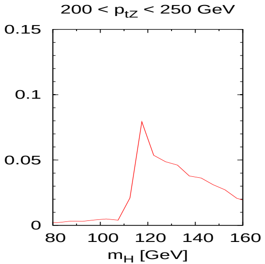
[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]

Butterworth, Davison, Rubin & GPS '08

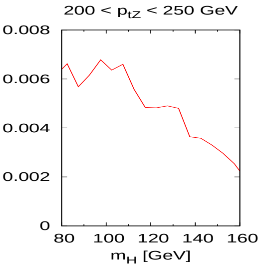
$pp \longrightarrow ZH \longrightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV



SIGNAL



Zbb BACKGROUND



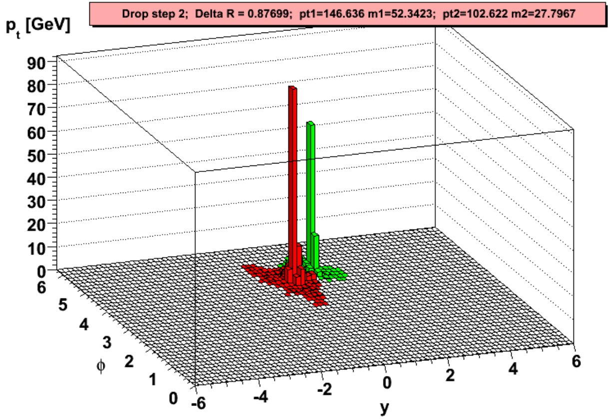
[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]

Butterworth, Davison, Rubin & GPS '08

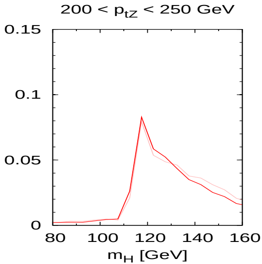
Possible new (light) Higgs discovery channel

arbitrary norm.

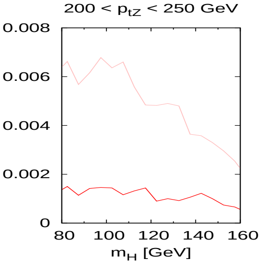
$pp \longrightarrow ZH \longrightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV



SIGNAL



Zbb BACKGROUND



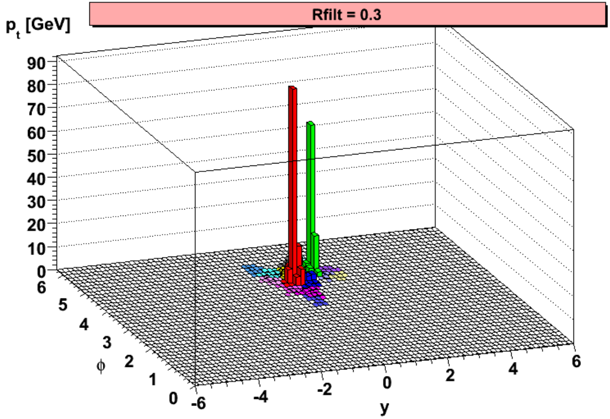
[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]

Butterworth, Davison, Rubin & GPS '08

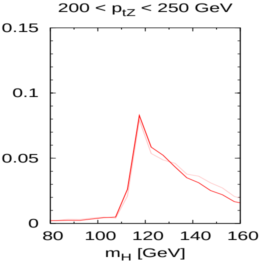
Possible new (light) Higgs discovery channel

arbitrary norm.

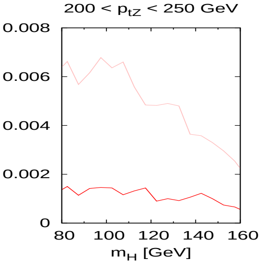
$pp \longrightarrow ZH \longrightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV



SIGNAL



Zbb BACKGROUND



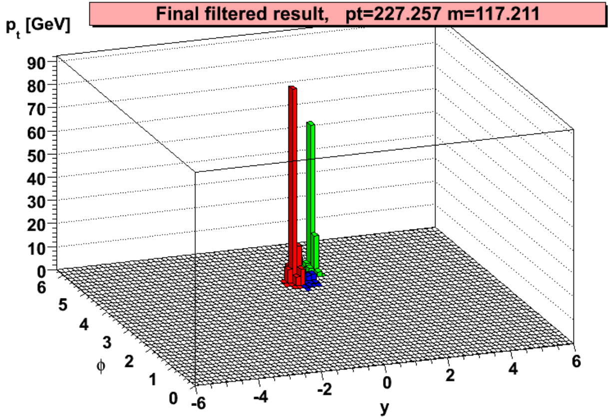
[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]

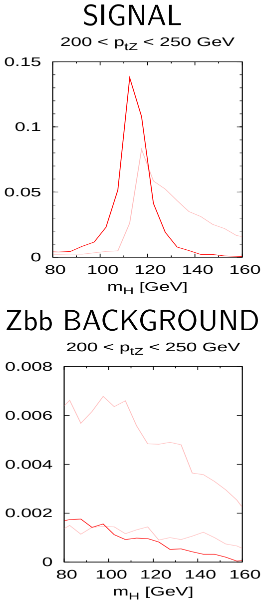
Butterworth, Davison, Rubin & GPS ’08

Possible new (light) Higgs discovery channel

arbitrary norm.

$pp \longrightarrow ZH \longrightarrow \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV





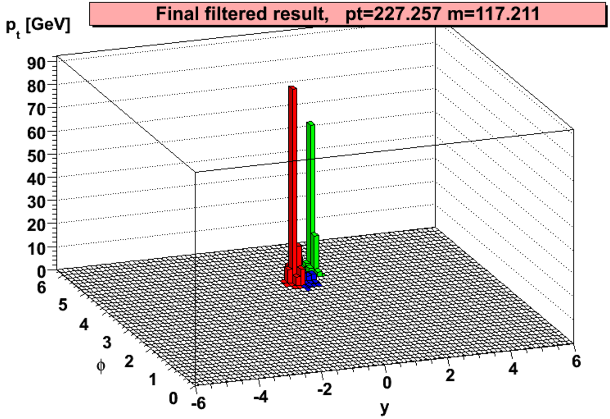
[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]

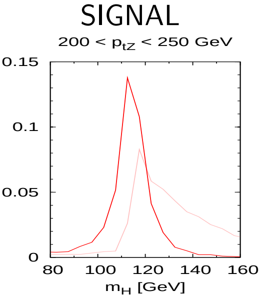
Butterworth, Davison, Rubin & GPS '08

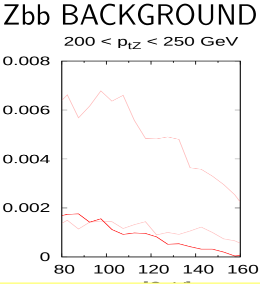
Possible new (light) Higgs discovery channel

arbitrary norm.

$pp \to ZH \to \nu\bar{\nu}b\bar{b}$, @14 TeV, $m_H = 115$ GeV







[Herwig 6.5 + Jimmy 4.31 + FastJet Cam/Aa R=1.2]

Butterworth, Davison, Rubin & GPS '08

Possible new (light) Higgs discovery channel

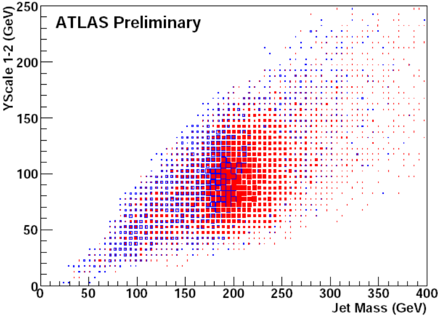
Much to be learnt still about extracting boosted W/H/Z/top?

High-p_t top $\longrightarrow b q \bar{q}'$

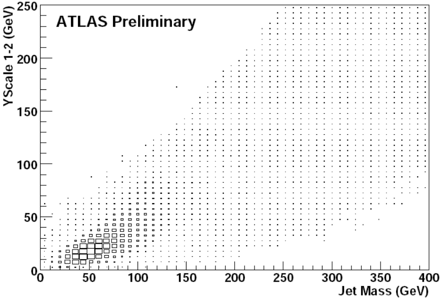
Brooijmans '08 ATL-PHYS-CONF-2008-008, based on k_t algorithm
+ Thaler & Wang '08; Almeida et al. '08 (k_t, jet-shapes)
+ Kaplan et al '08 (C/A decomposition)

Use subjet relative transverse-momentum scale ("'y-scale") & correlation with jet mass to pick out top quarks from background

top quarks $p_t \sim 1$ TeV



normal jets

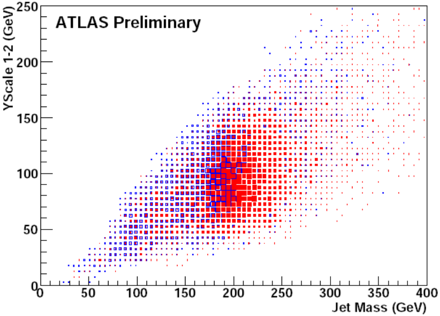


High-p_t top $\longrightarrow b q \bar{q}'$

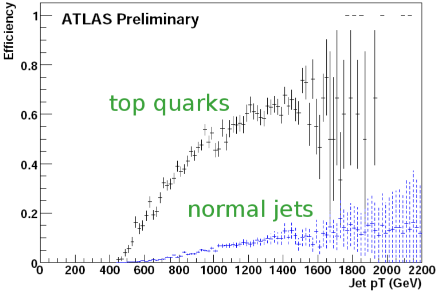
Brooijmans '08 ATL-PHYS-CONF-2008-008, based on k_t algorithm
+ Thaler & Wang '08; Almeida et al. '08 (k_t, jet-shapes)
+ Kaplan et al '08 (C/A decomposition)

Use subjet relative transverse-momentum scale ("'y-scale") & correlation with jet mass to pick out top quarks from background

top quarks $p_t \sim 1$ TeV



efficiencies



Conclusions

Conclusions

- A jet is not a parton: it's (sort of) what you choose it to be.
- It's easier to think in terms of partons (LO, NLO pQCD) with IR/Collinear safe jet algorithms. And gives sense to pQCD predictions
- ∃ many cones algs. Not equivalent. Many are IR/Coll unsafe.
  xC-SM → SISCone; xC-PR → anti-k_t
- "The best" jet definition *does not exist*
- To get the most out of jet-algs.,
  - Understand the interplay of physical scales high $p_t \rightarrow$ larger R
  - Try out different combinations of algorithm & R
  - Check Variations of alg. & R don't change extracted physical quantities
- Special cases (e.g. boosted W/t/...) benefit from special techniques
  e.g. seq. recomb. "jet-decomposition" is a powerful tool