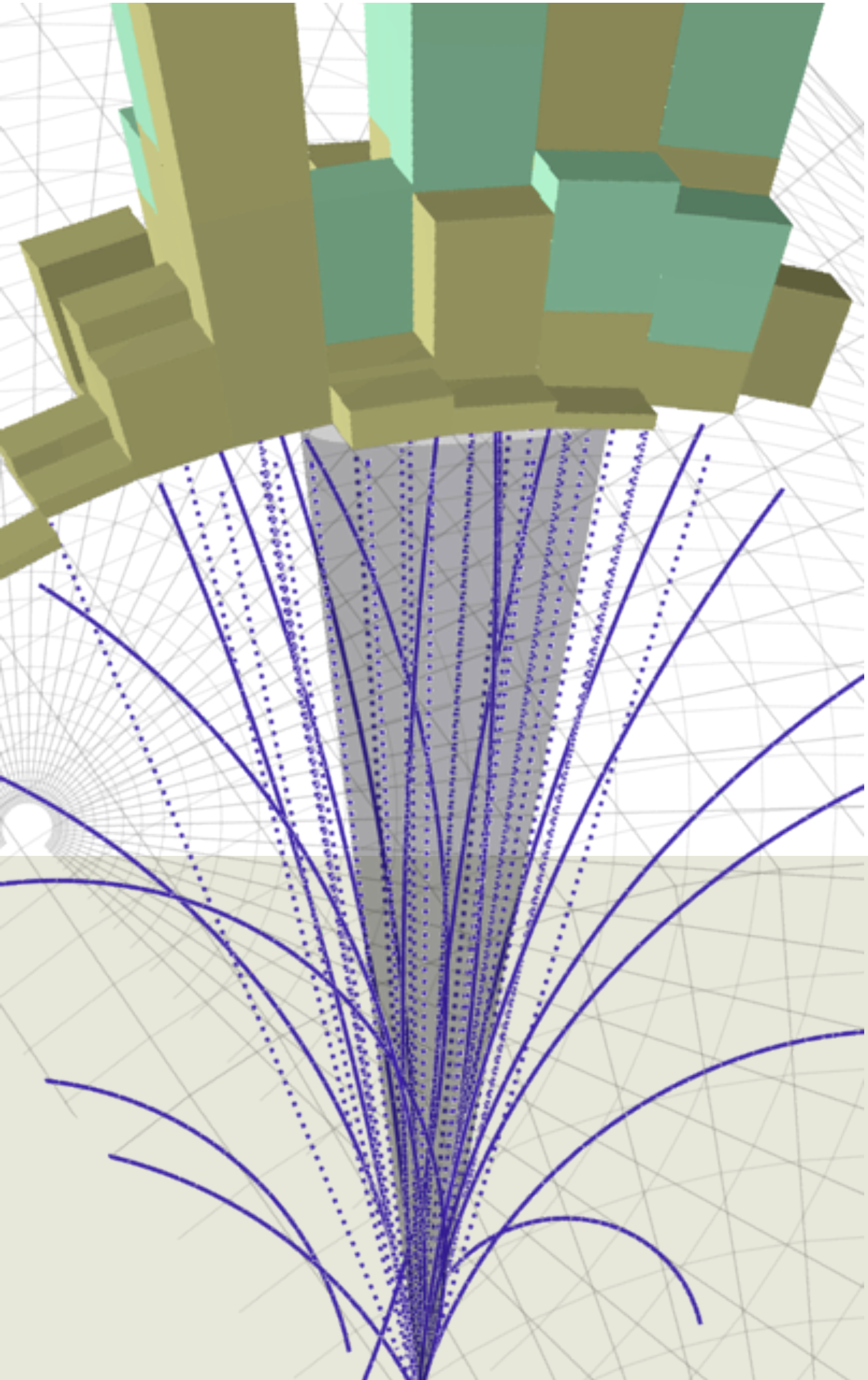
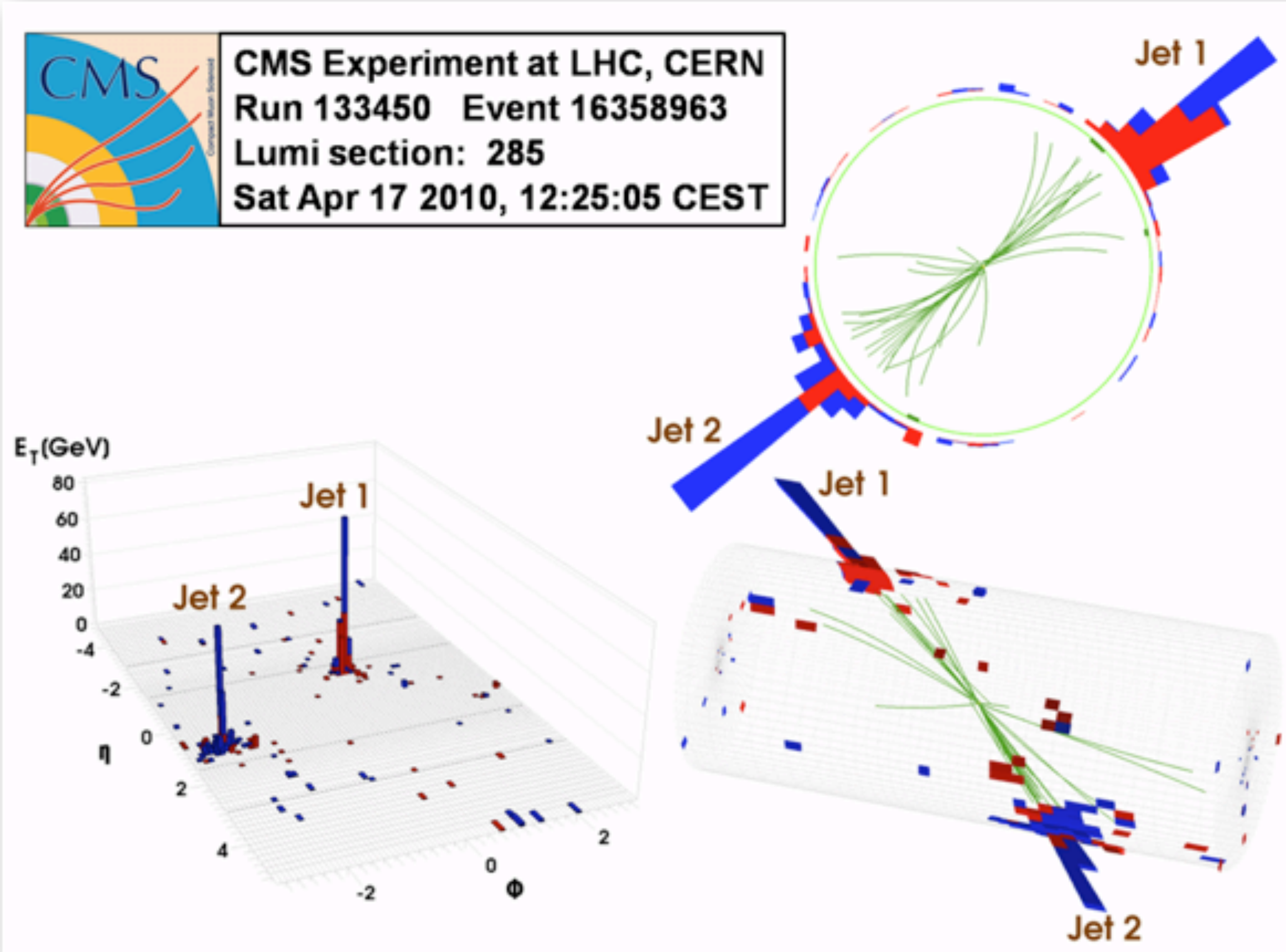
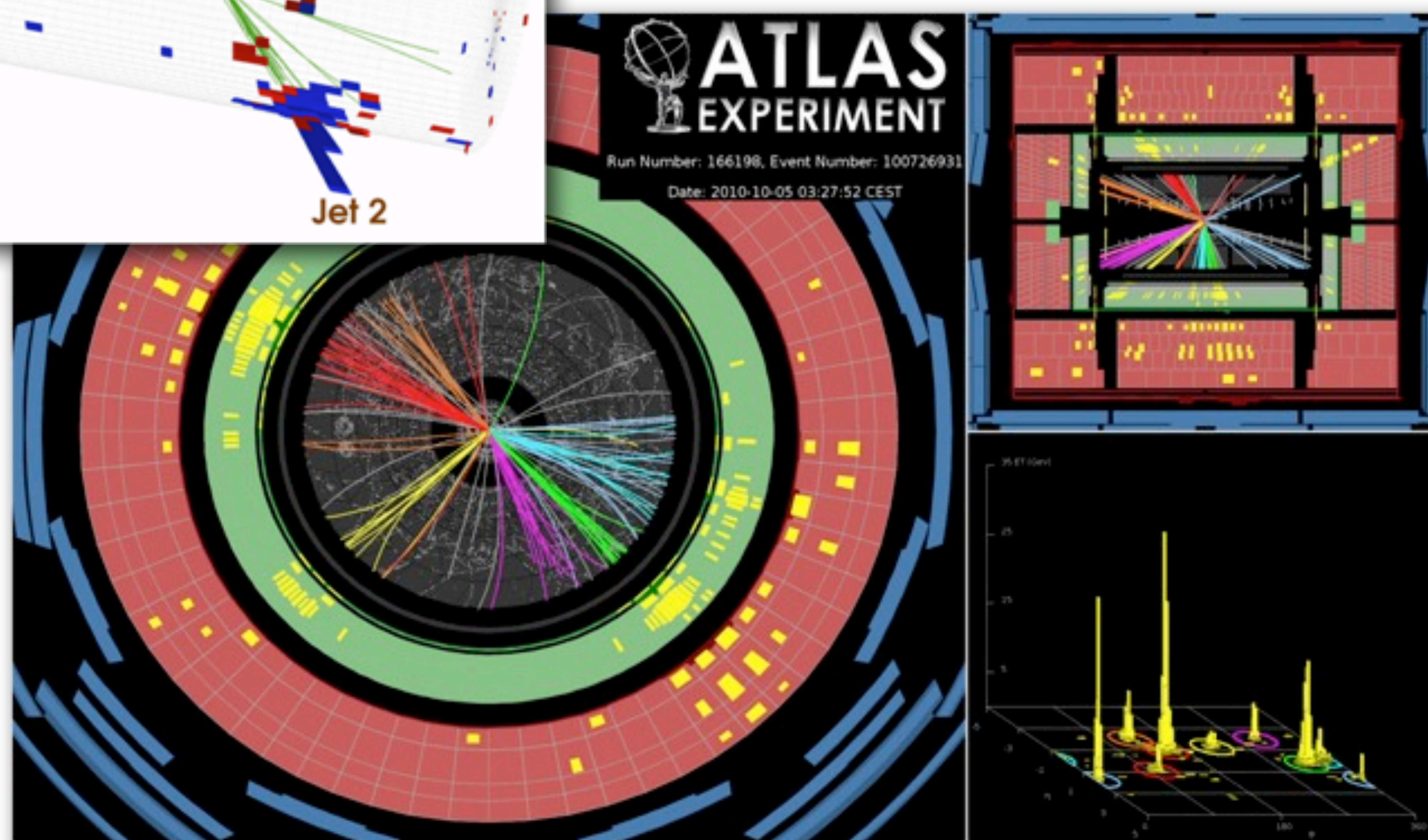


Jets and jet substructure

Gavin Salam (CERN)
with extensive use of material by Matteo Cacciari and Gregory Soyez

CFHEP
April 2014

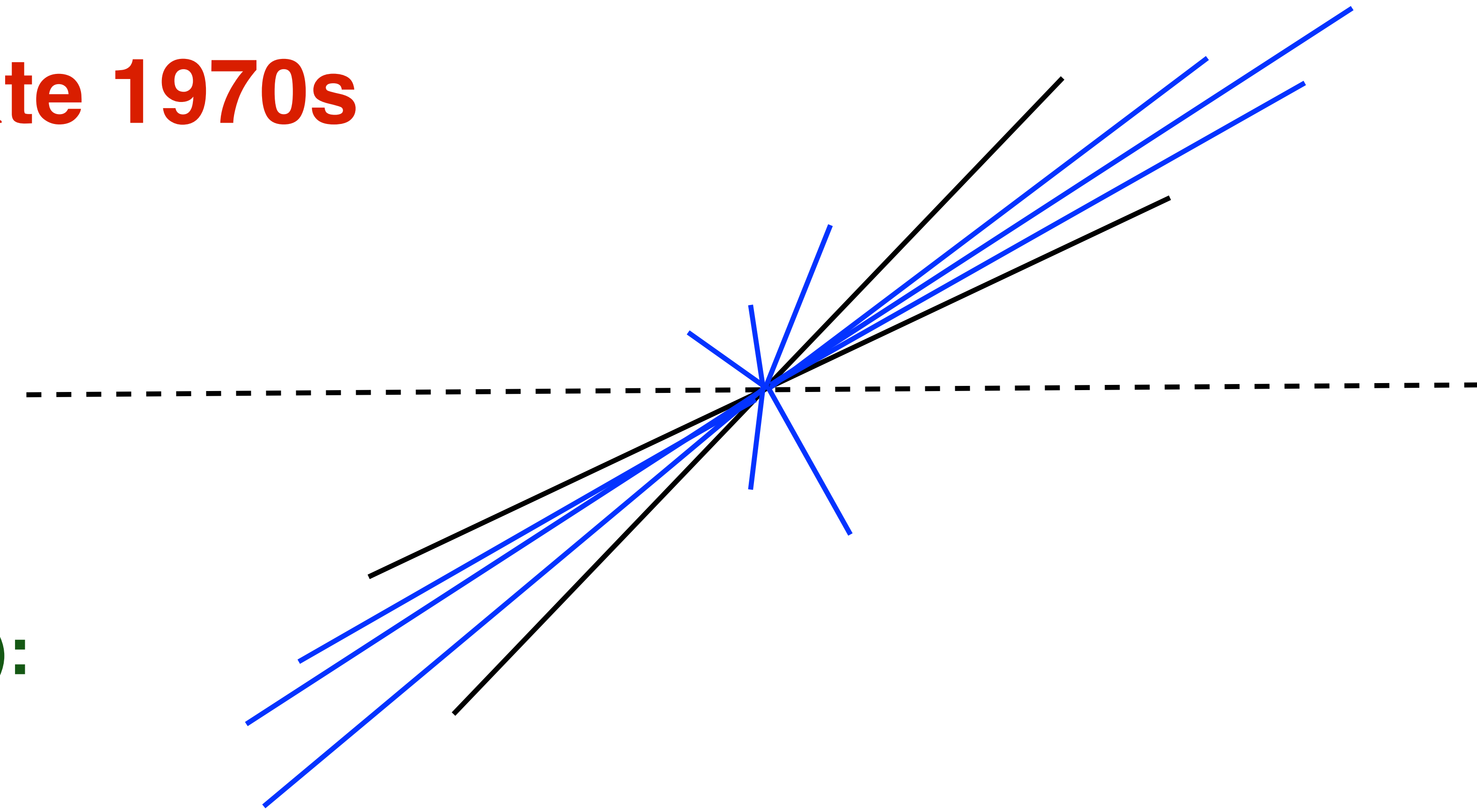




JETS

Collimated, energetic bunches of particles

Jets date back to the late 1970s

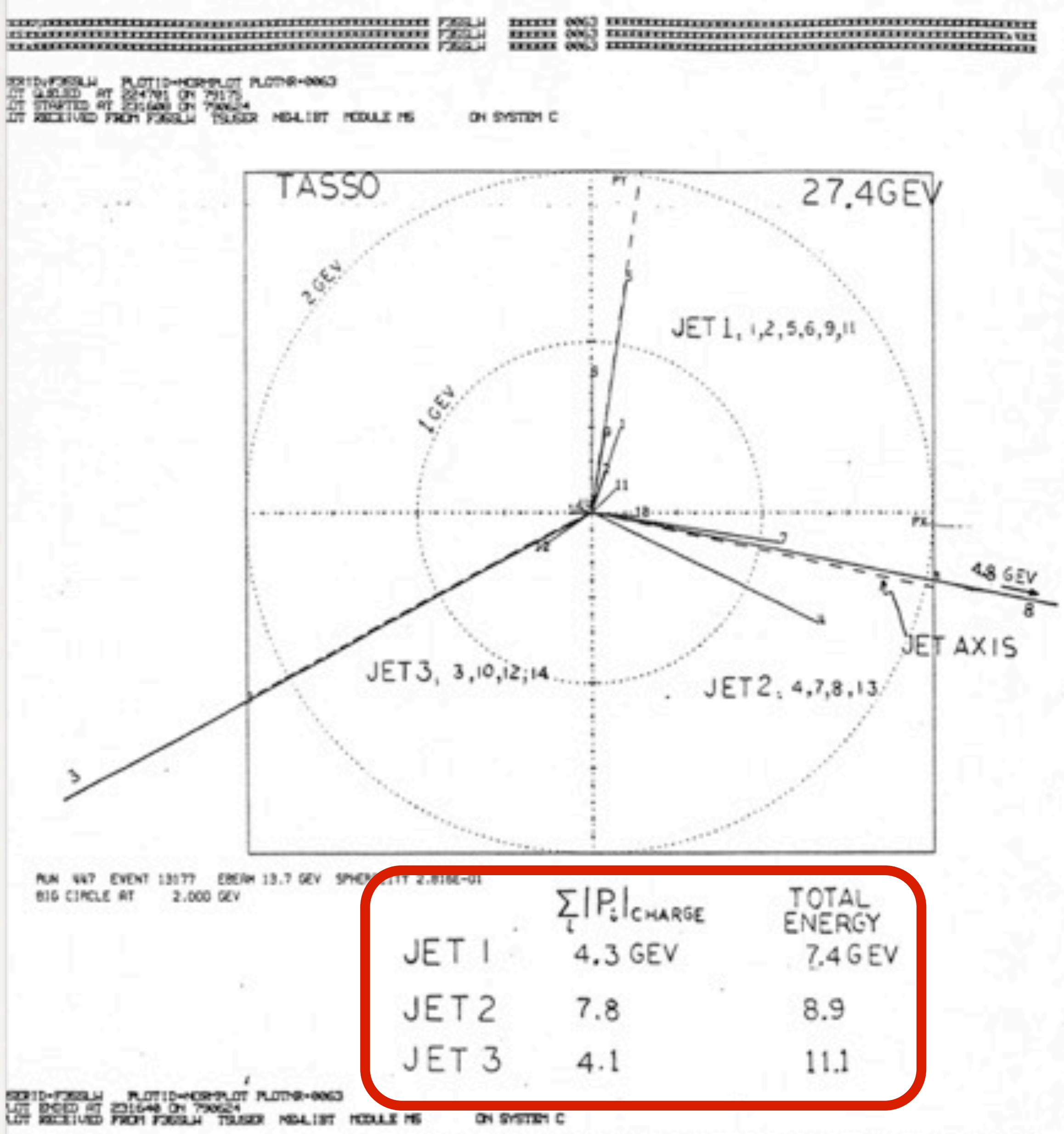


Sterman and Weinberg,
Phys. Rev. Lett. 39, 1436 (1977):

To study jets, we consider the partial cross section $\sigma(E,\theta,\Omega,\varepsilon,\delta)$ for e^+e^- hadron production events, in which all but a fraction $\varepsilon \ll 1$ of the total e^+e^- energy E is emitted within some pair of oppositely directed cones of half-angle $\delta \ll 1$, lying within two fixed cones of solid angle Ω (with $\pi\delta^2 \ll \Omega \ll 1$) at an angle θ to the e^+e^- beam line. We expect this to be measur-

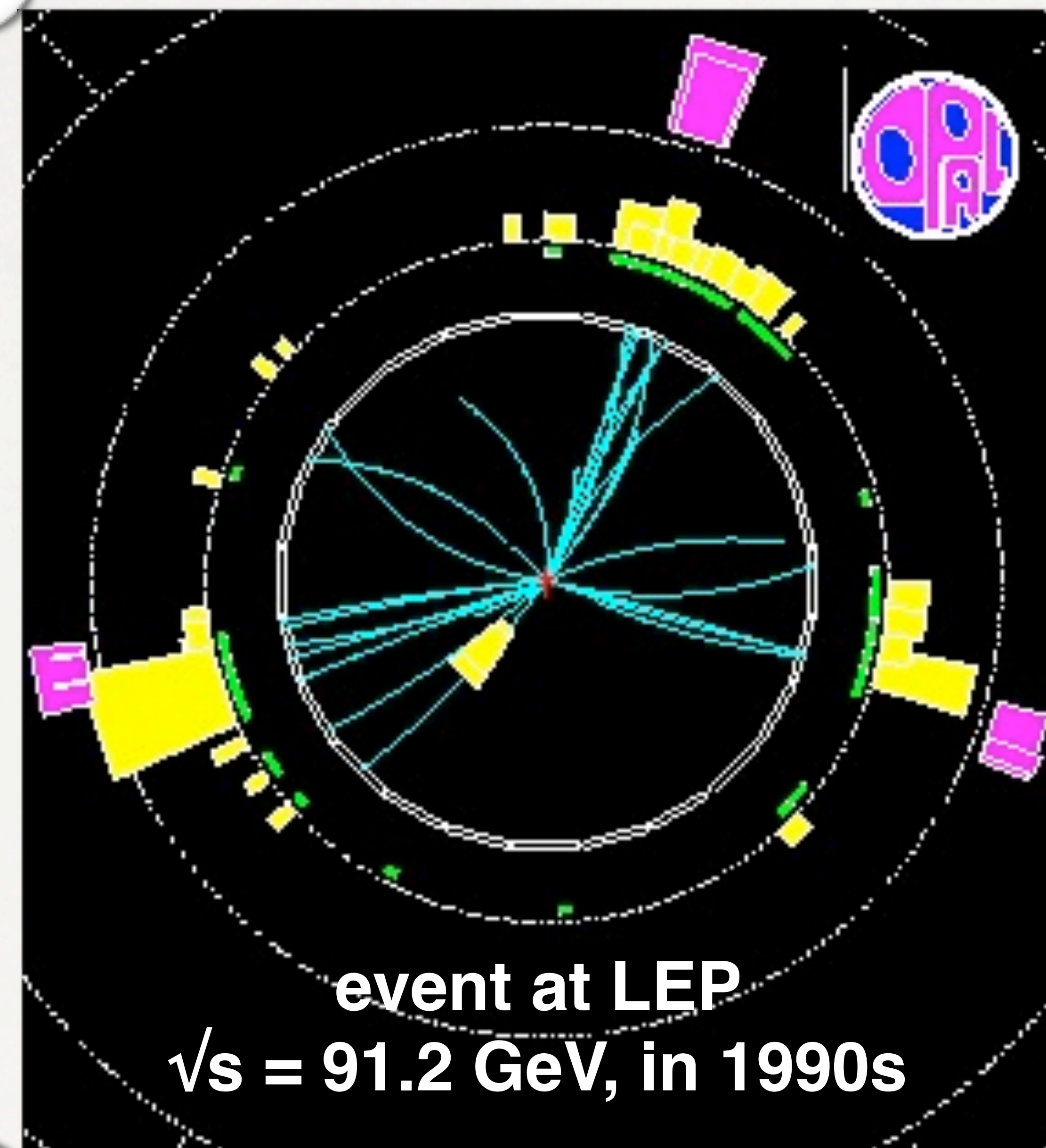
$$\sigma(E,\theta,\Omega,\varepsilon,\delta) = (d\sigma/d\Omega)_0\,\Omega\left[1 - (g_E^2/3\pi^2)\left\{3\ln\delta + 4\ln\delta\,\ln 2\varepsilon + \frac{\pi^3}{3} - \frac{5}{2}\right\}\right]$$

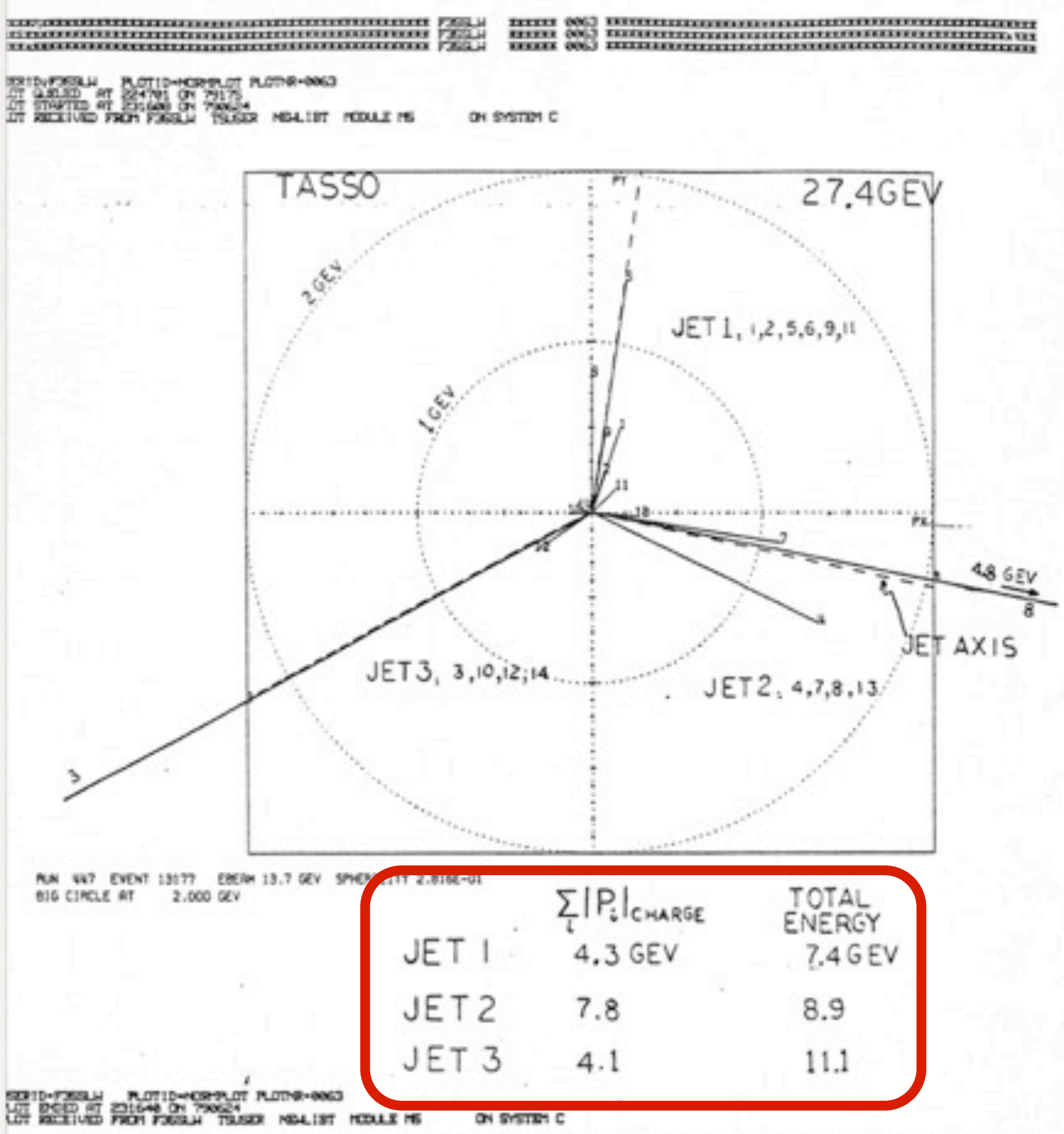
And they've been used and studied at every collider since



gluon discovery:

event at TASSO
$\sqrt{s}$ = 27.4 GeV, in 1979

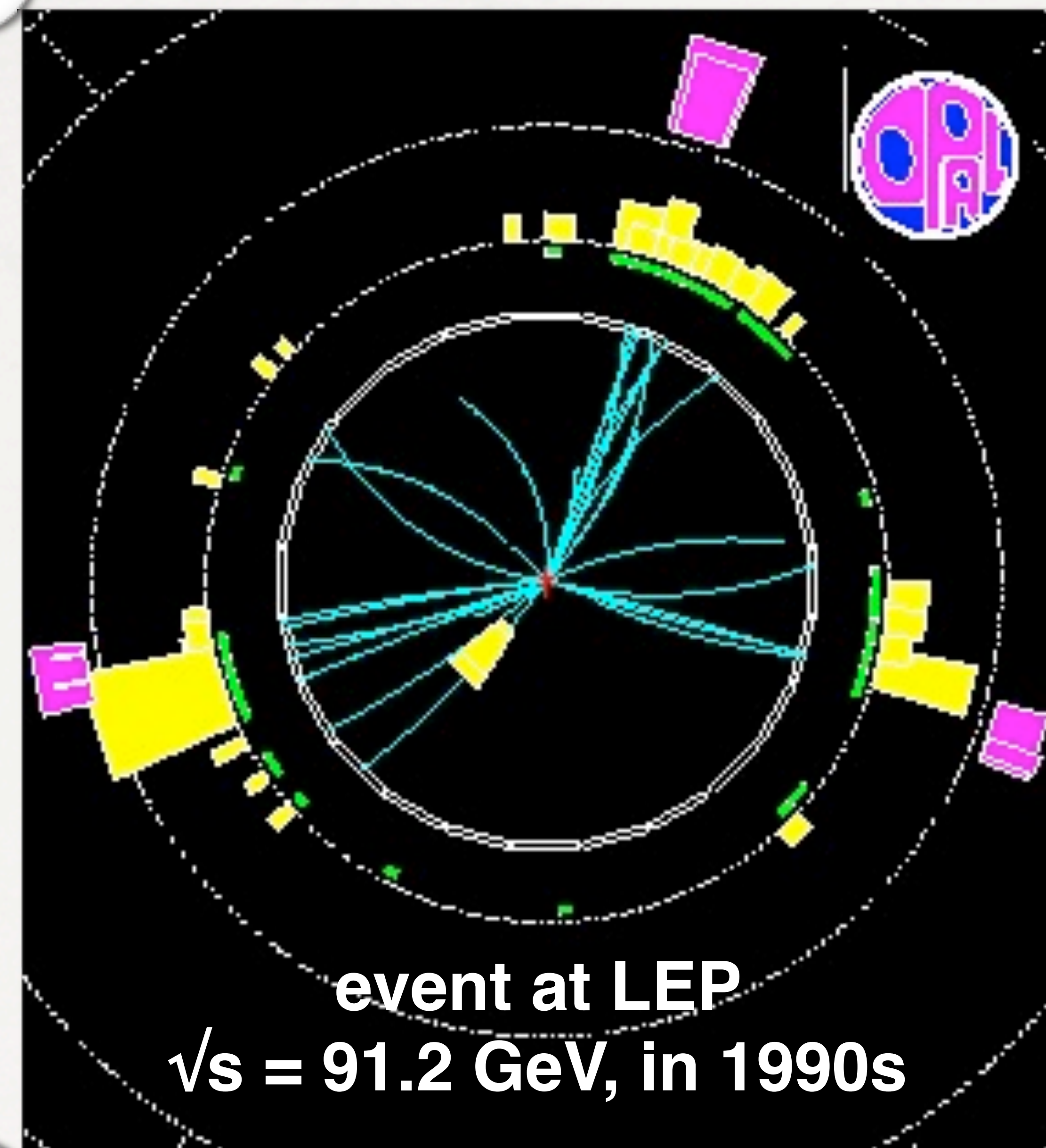




gluon discovery:

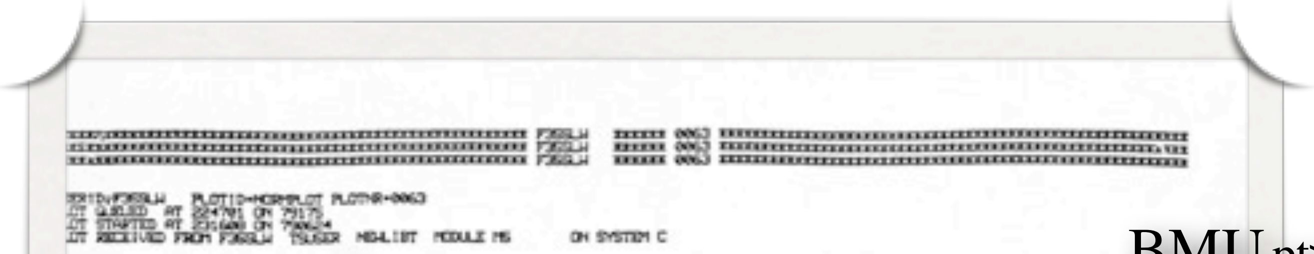
event at TASSO
$\sqrt{s}$ = 27.4 GeV, in 1979

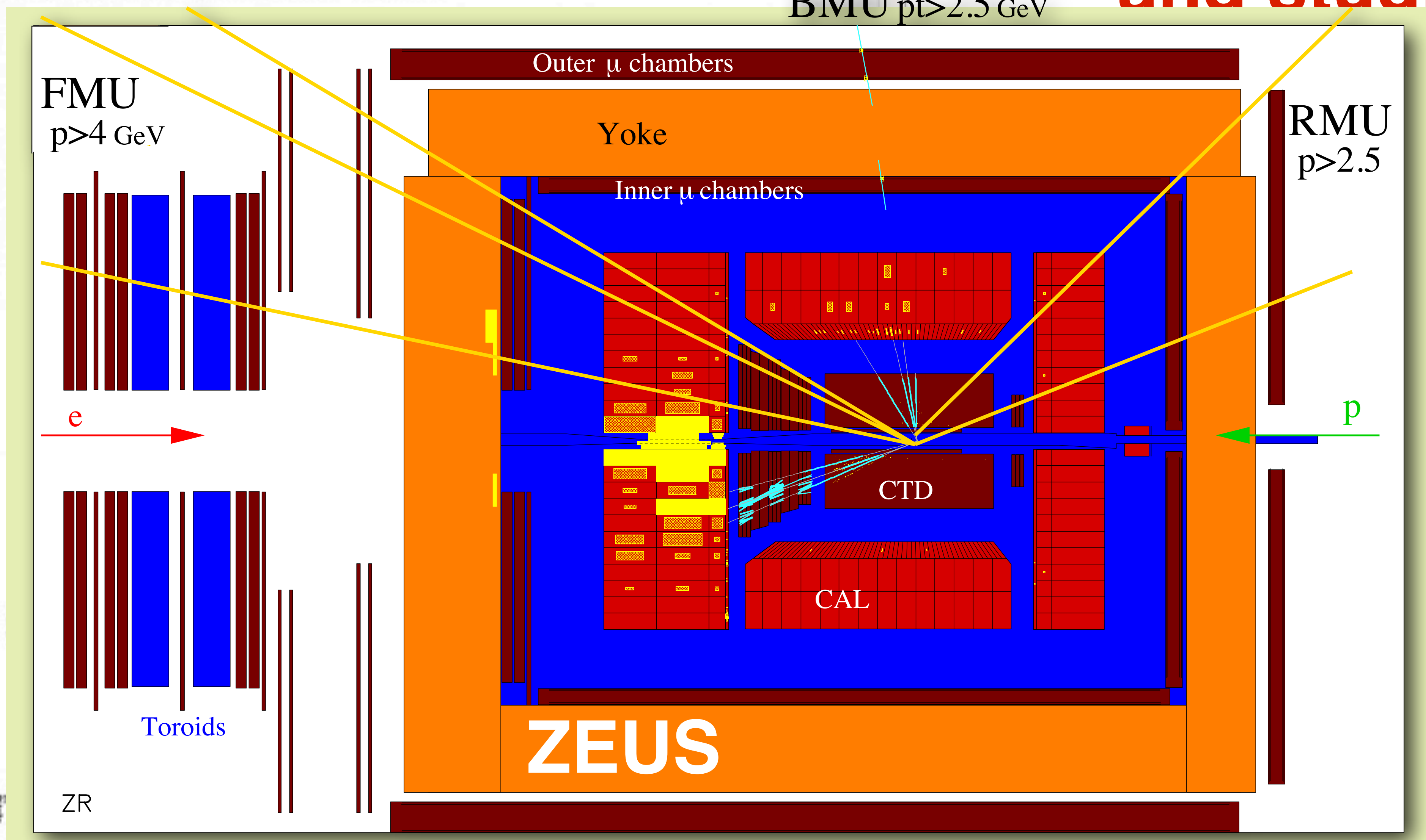
And they've been used and studied at every collider since



event at LEP
$\sqrt{s}$ = 91.2 GeV, in 1990s

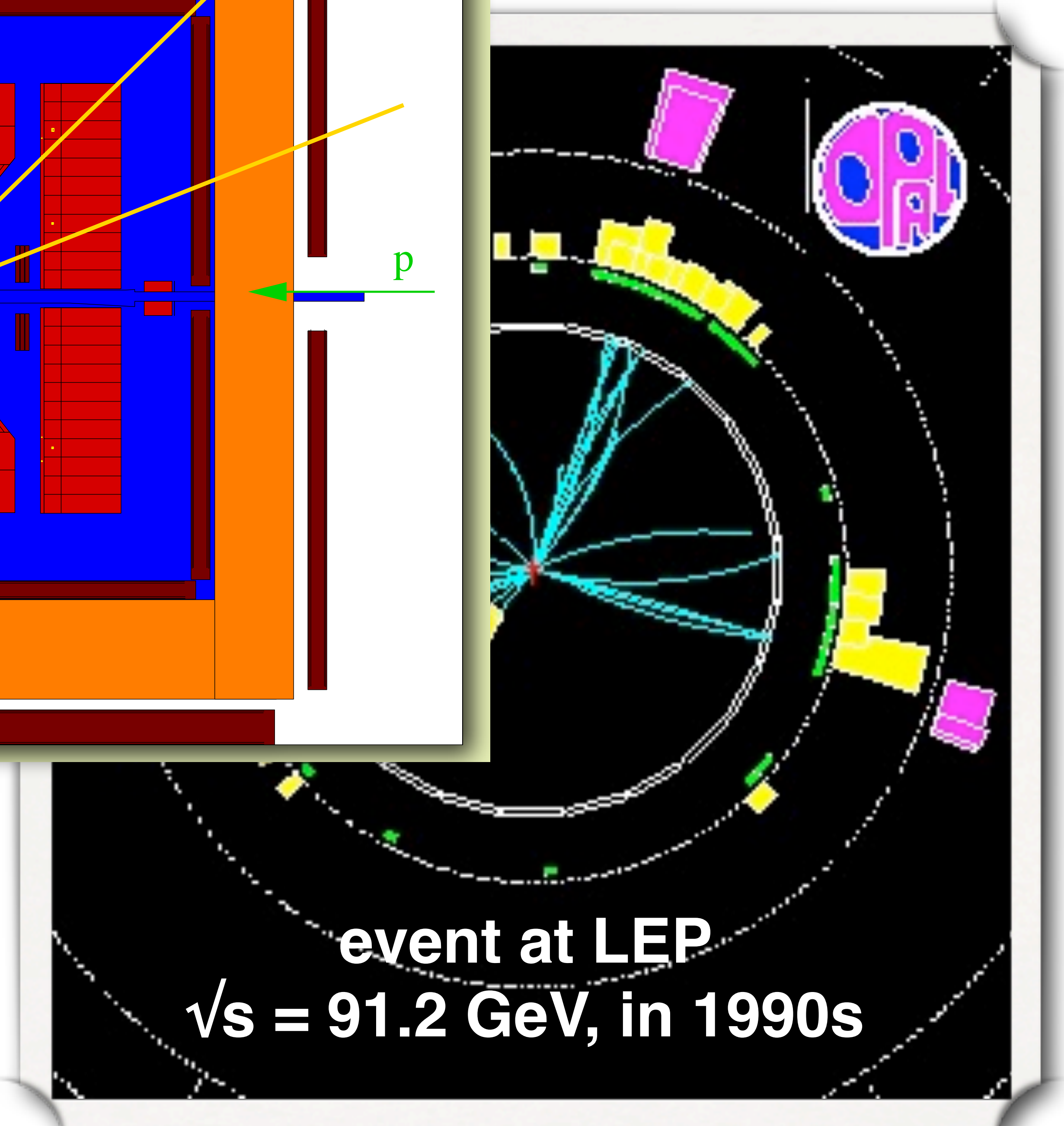
And they've been used and studied at every ... er since





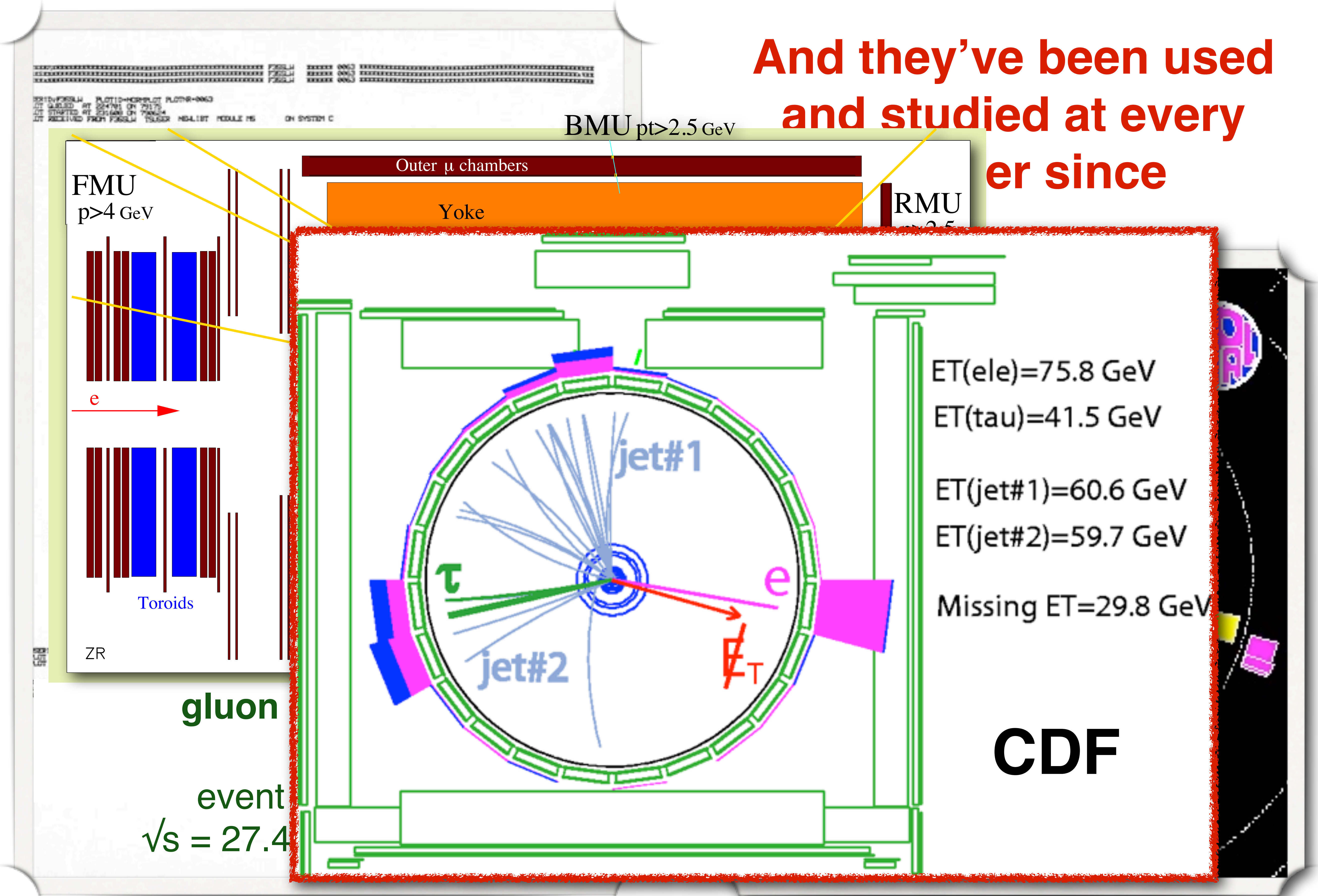
gluon discovery:

event at TASSO
√s = 27.4 GeV, in 1979



event at LEP
√s = 91.2 GeV, in 1990s

And they've been used and studied at every ... er since



Find all papers by ATLAS and CMS
578 records found

reportnumber:CERN-PH-EP (collaboration:atlas or collaboration:CMS) | Brief format | Search | Easy Search | Advanced Search

find j "Phys.Rev.Lett.,105*" :: more

Sort by: latest first | desc. | - or rank by -

Display results: 25 results | single list

No exact match found for *cern-ph-ep*, using *cern ph ep* instead...

HEP **464** records found 1 - 25 jump to record: 1 Search took 0.04 seconds.

1. **Measurement of the W+W- cross section in pp collisions at sqrt(s) = 7 TeV and limits on anomalous WW gamma and WWZ couplings**
   CMS Collaboration (Serguei Chatrchyan (Yerevan Phys. Inst.) *et al.*). Jun 5, 2013.
   CMS-SMP-12-005, CERN-PH-EP-2013-075
   e-Print: **arXiv:1306.1126 [hep-ex]** | **PDF**
   References | BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote
   ADS Abstract Service
   Detailed record

2. **Measurement of the hadronic activity in events with a Z and two jets and extraction of the cross section for the electroweak production of a Z with two jets in pp collisions at $\sqrt{s}$ = 7 TeV**
   CMS Collaboration (Serguei Chatrchyan (Yerevan Phys. Inst.) *et al.*). May 31, 2013.
   CMS-FSQ-12-019, CERN-PH-EP-2013-060
   e-Print: **arXiv:1305.7389 [hep-ex]** | PDF
   References | BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote
   ADS Abstract Service
   Detailed record

Pull out those that refer to one widely used jet-alg

347 records found

reportnumber:CERN-PH-EP (collaboration:atlas or collaboration:CMS) (refersto:recid:779080) Brief format Search Easy Search Advanced Search

find j "Phys.Rev.Lett.,105*" :: more

Sort by: latest first | desc. | - or rank by -

Display results: 25 results | single list

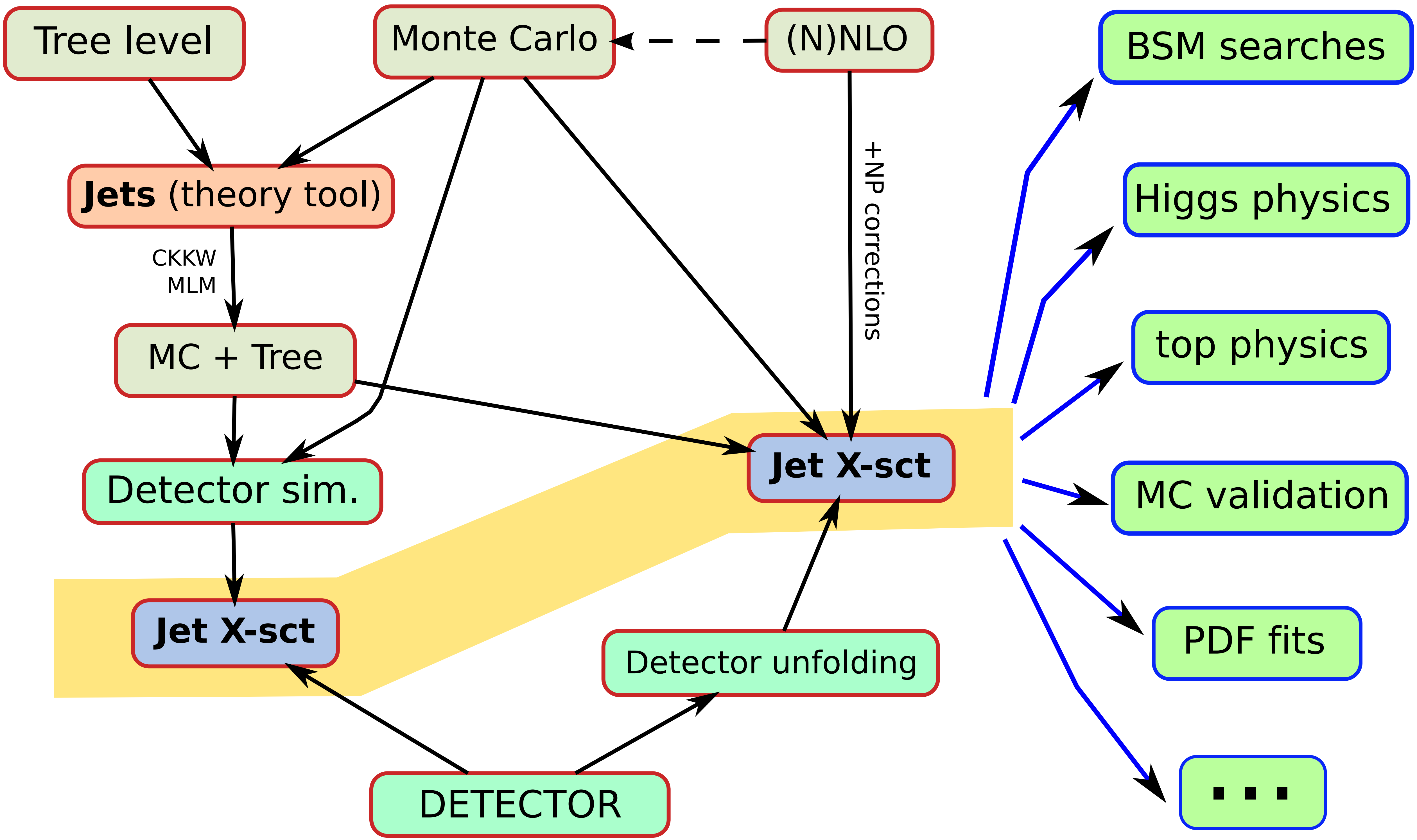
60% of papers use jets!

No exact match found for *cern-ph-ep*, using *cern ph ep* instead...

HEP **272** records found 1 - 25 jump to record: 1 Search took 0.04 seconds.

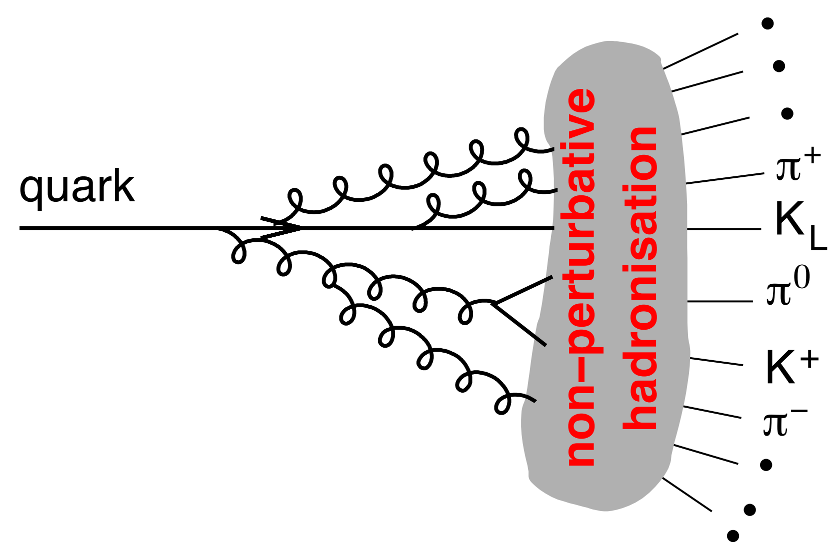
1. **Measurement of the W+W- cross section in pp collisions at sqrt(s) = 7 TeV and limits on anomalous WW gamma and WWZ couplings**
CMS Collaboration (Serguei Chatrchyan (Yerevan Phys. Inst.) *et al.*). Jun 5, 2013.
CMS-SMP-12-005, CERN-PH-EP-2013-075
e-Print: **arXiv:1306.1126 [hep-ex]** | **PDF**
References | BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote
ADS Abstract Service
Detailed record

2. **Measurement of the hadronic activity in events with a Z and two jets and extraction of the cross section for the electroweak production of a Z with two jets in pp collisions at $\sqrt{s}$ = 7 TeV**
CMS Collaboration (Serguei Chatrchyan (Yerevan Phys. Inst.) *et al.*). May 31, 2013.
CMS-FSQ-12-019, CERN-PH-EP-2013-060
e-Print: **arXiv:1305.7389 [hep-ex]** | PDF
References | BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote
ADS Abstract Service
Detailed record

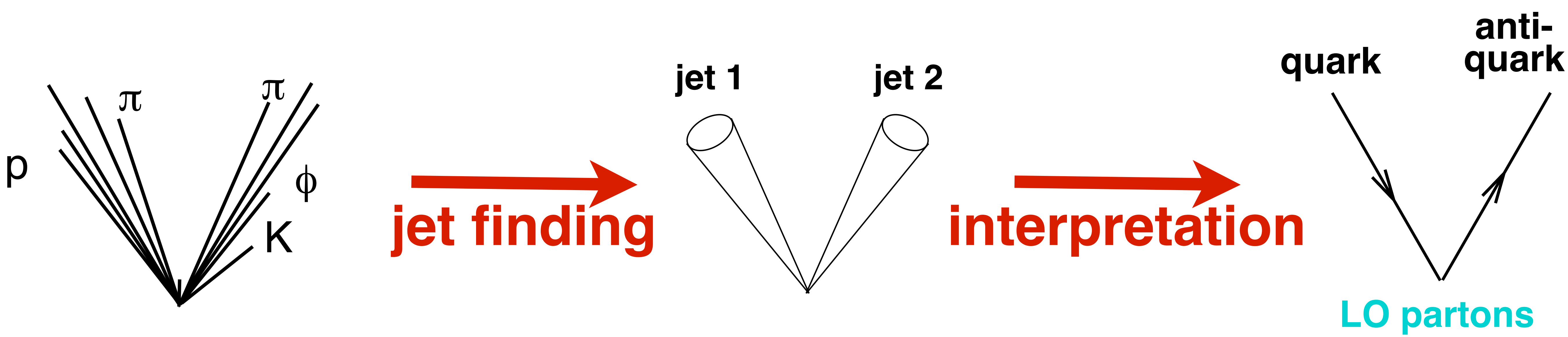


Tree level
Monte Carlo
(N)NLO
BSM searches
Jets (theory tool)
Higgs physics
+NP corrections
CKKW
MLM
MC + Tree
top physics
Jet X-sct
Detector sim.
MC validation
Jet X-sct
PDF fits
Detector unfolding
...
DETECTOR

How jets come to have the structure they do

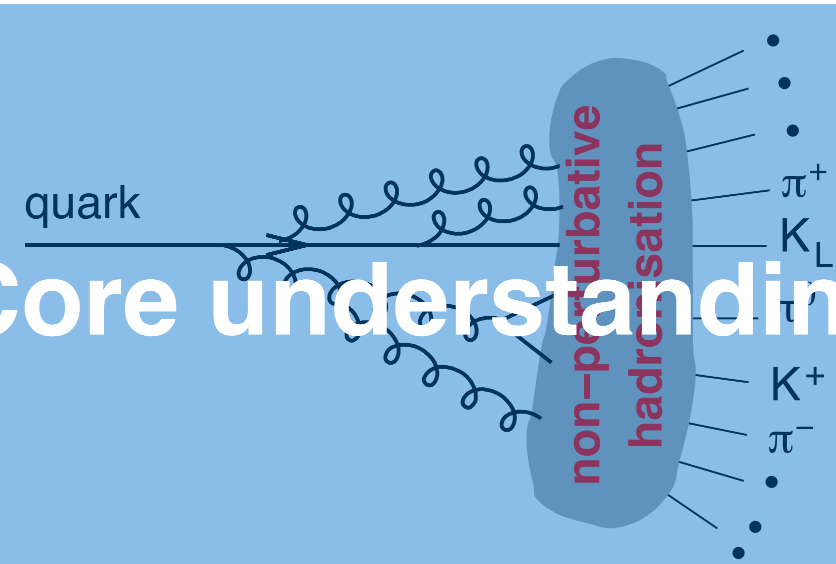


How we "reconstruct" jets



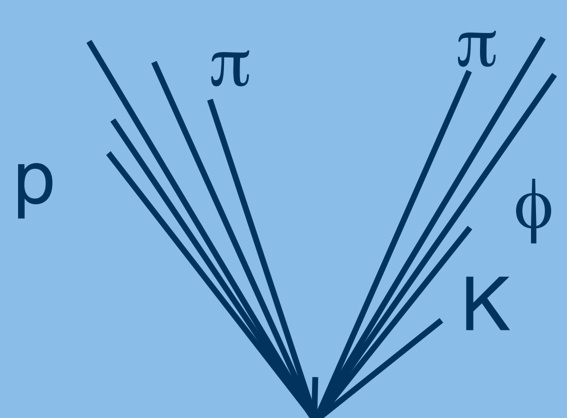
Two key aspects to discussing jets

How jets come to have the structure they do

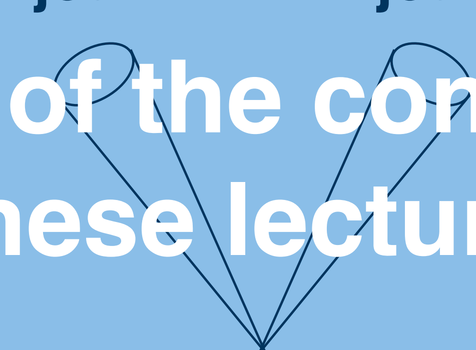


Core understanding

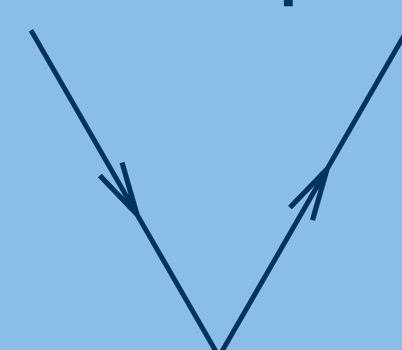
How we "reconstruct" jets



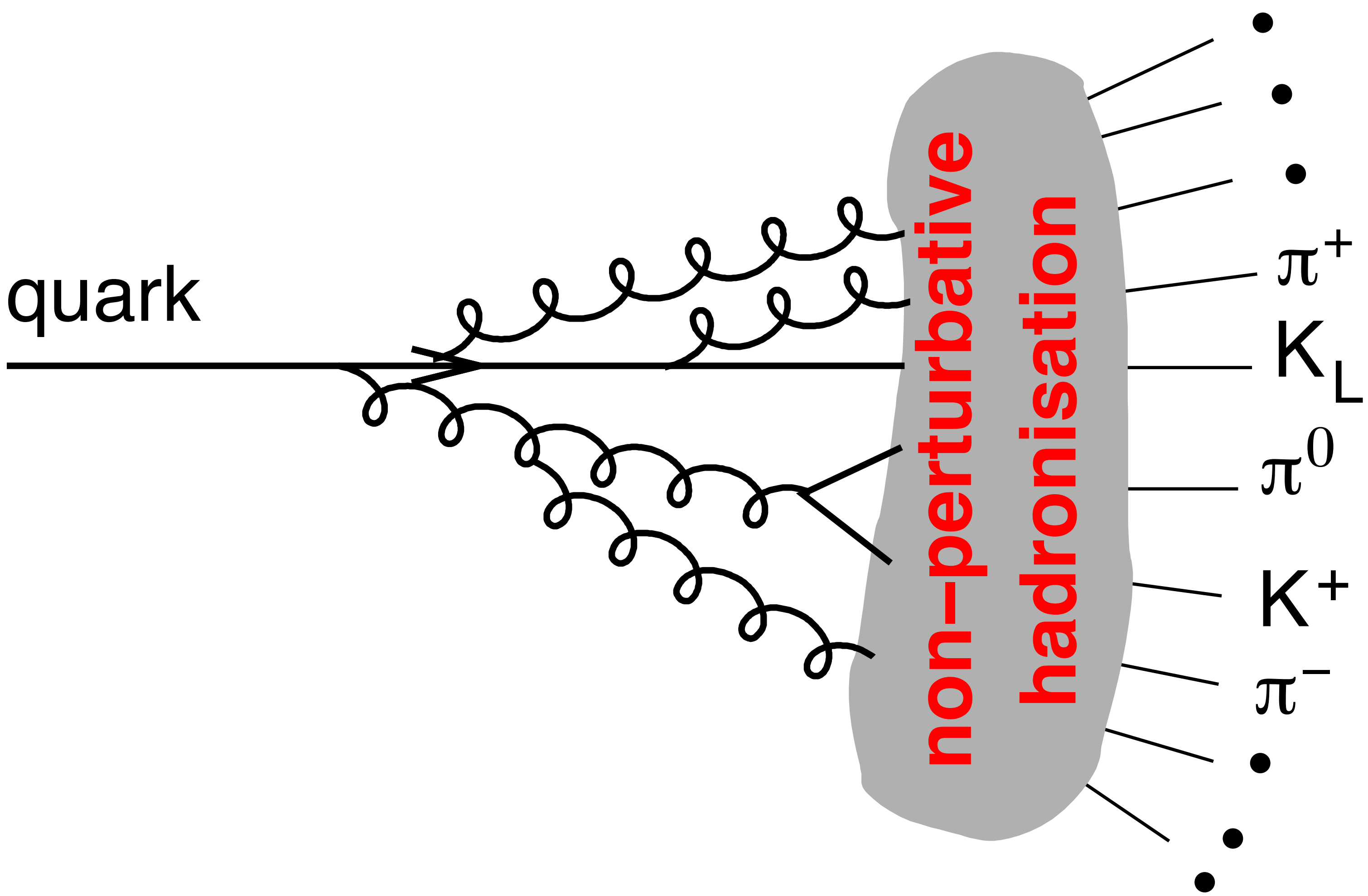
jet finding →



→ **interpretation**



Most of the content of these lectures



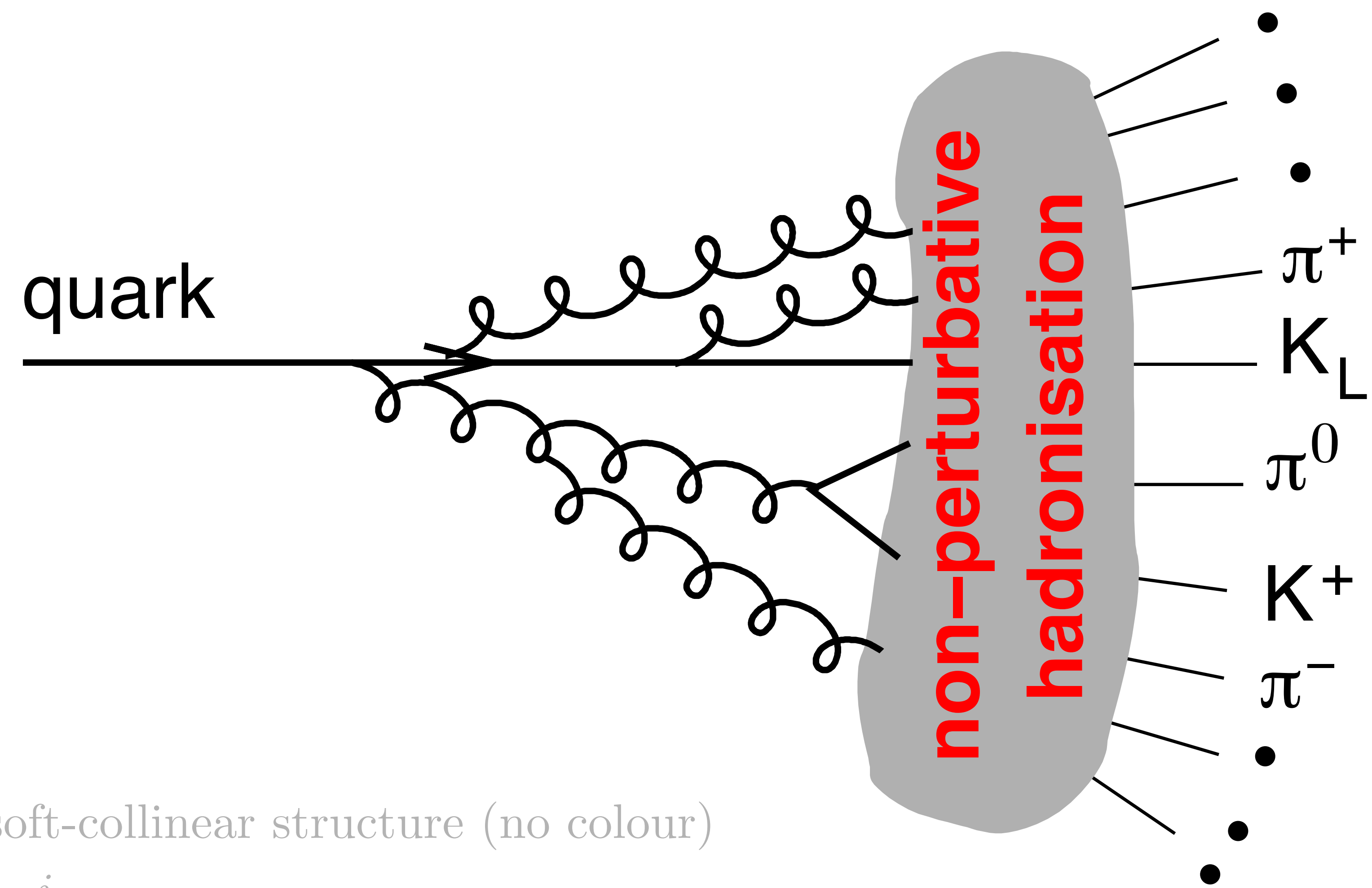
Gluon emission

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

Non-perturbative physics

$$\alpha_s \sim 1$$

Why do we see jets?



Gluon emission

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

Non-perturbative physics

$$\alpha_s \sim 1$$

soft-collinear structure (no colour)

$$\frac{i}{\not p + \not k}\, i g_s \not\epsilon\, v(p)$$

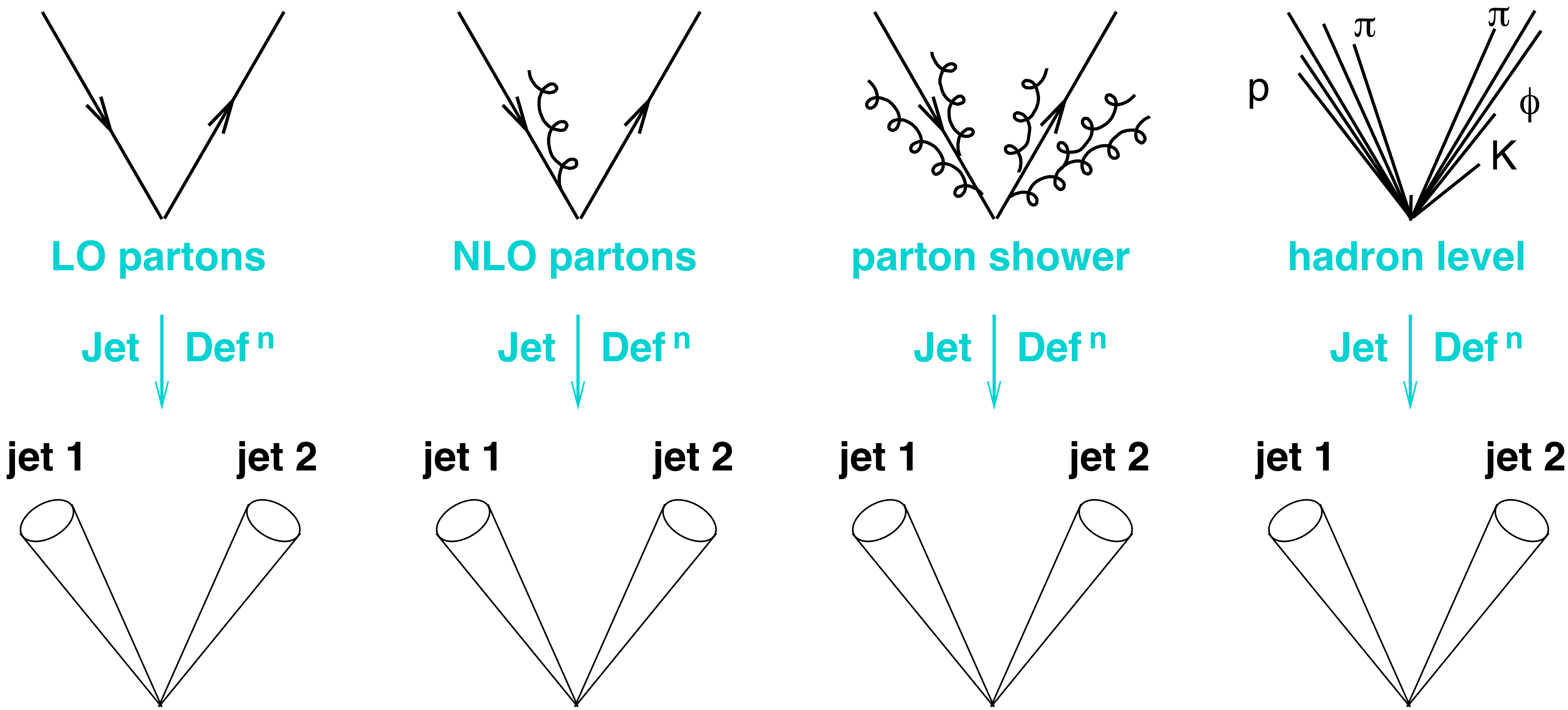
drop $\not k$ in numerator, use $\not p \not k + \not k \not p = 2p.k$, $\not p v(p) = 0$

get $g_s \dfrac{p.\epsilon}{p.k} v(p)$ take $p = (1,0,0,1)$, $k = E(\cos\theta, \sin\theta, 0, 1)$, $\epsilon = (\sin\theta, -\cos\theta, 0, 0)$

square the amplitude, put in phasespace: $\dfrac{d^3k}{2E(2\pi)^3} \to \dfrac{E\, dE\, d\cos\theta\, d\phi}{16\pi^3}$

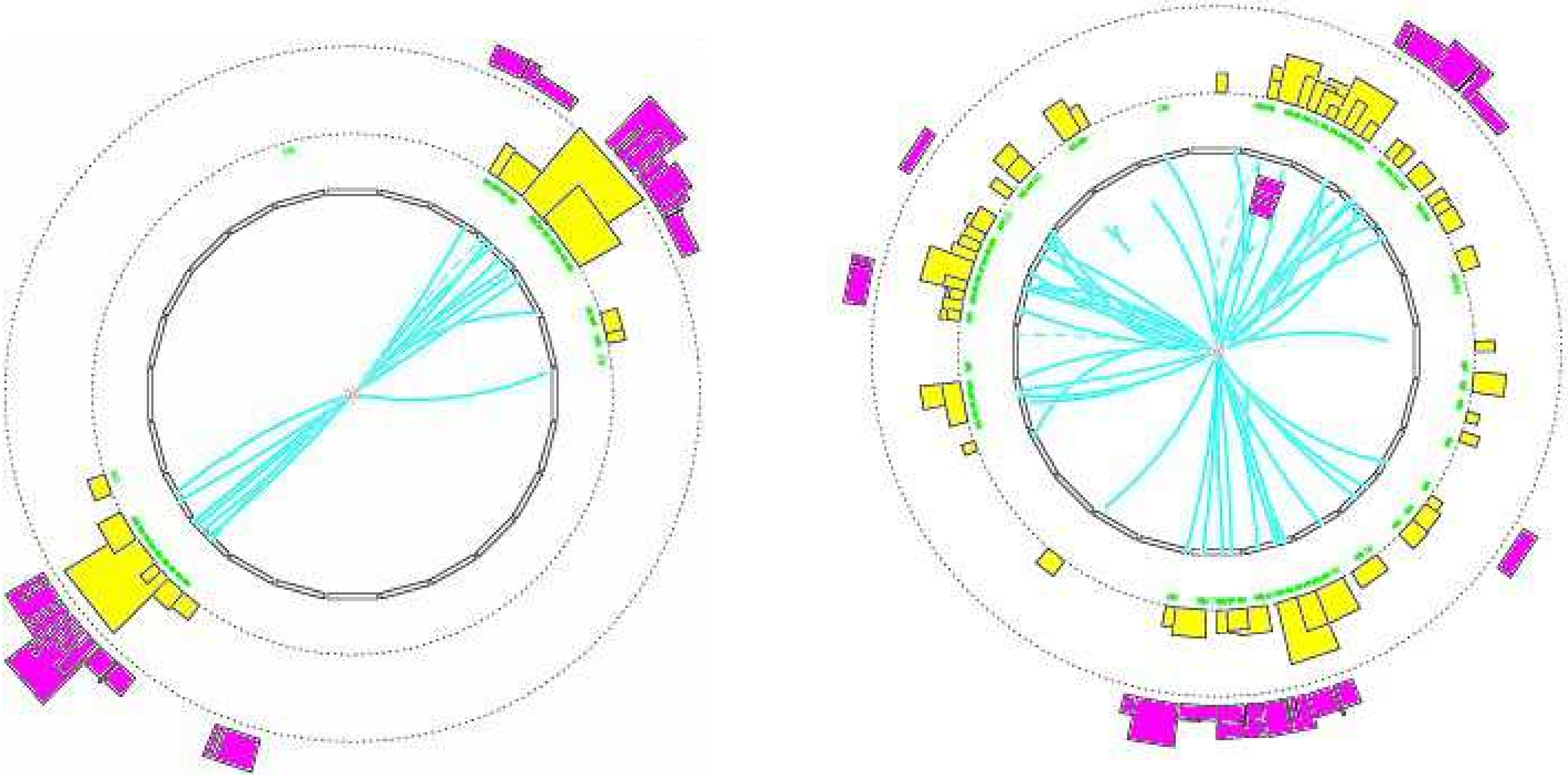
Reconstructing jets

Jet finding as a form of projection

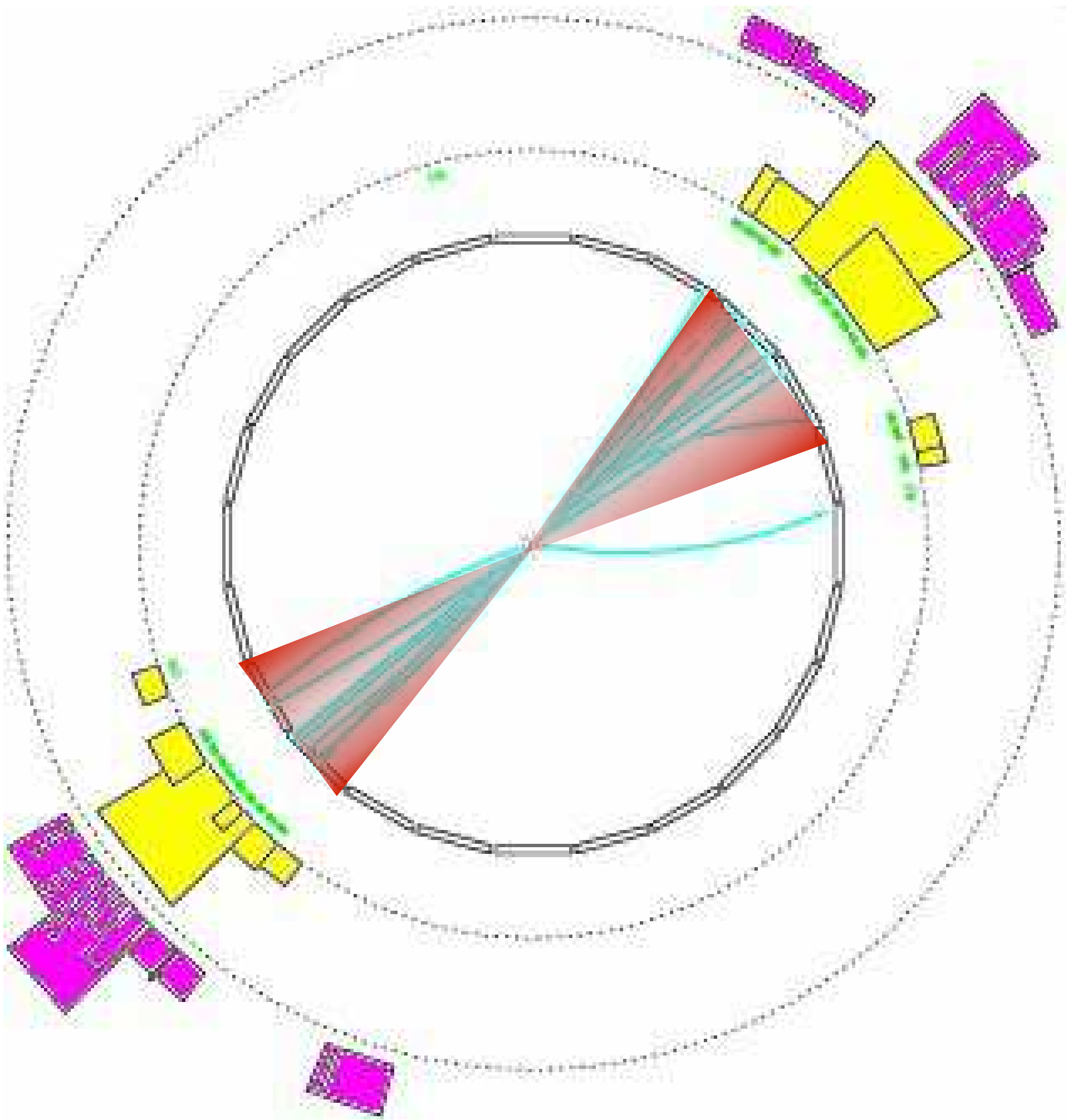


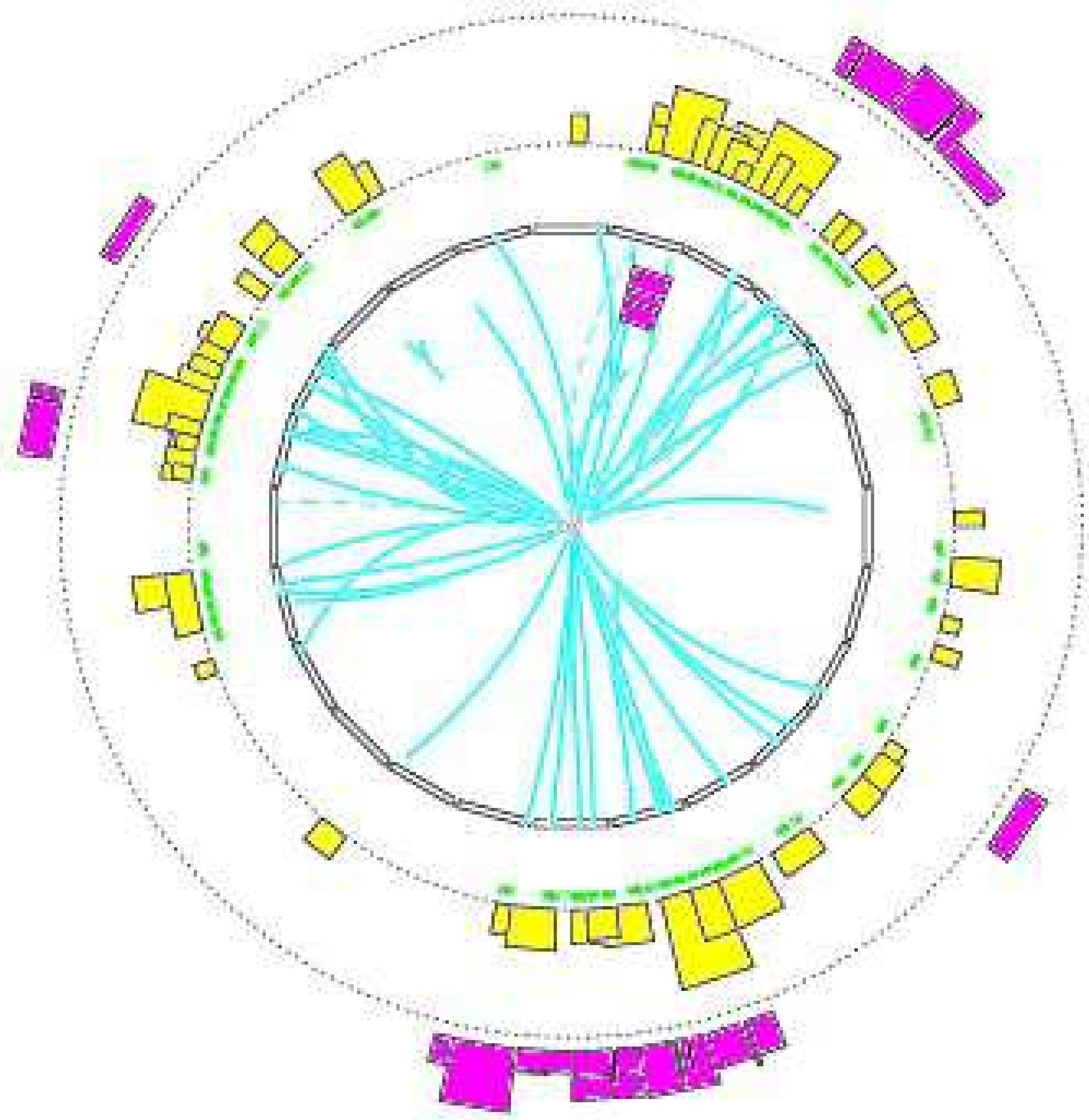
Projection to jets should be resilient to QCD effects

Reconstructing jets is an ambiguous task



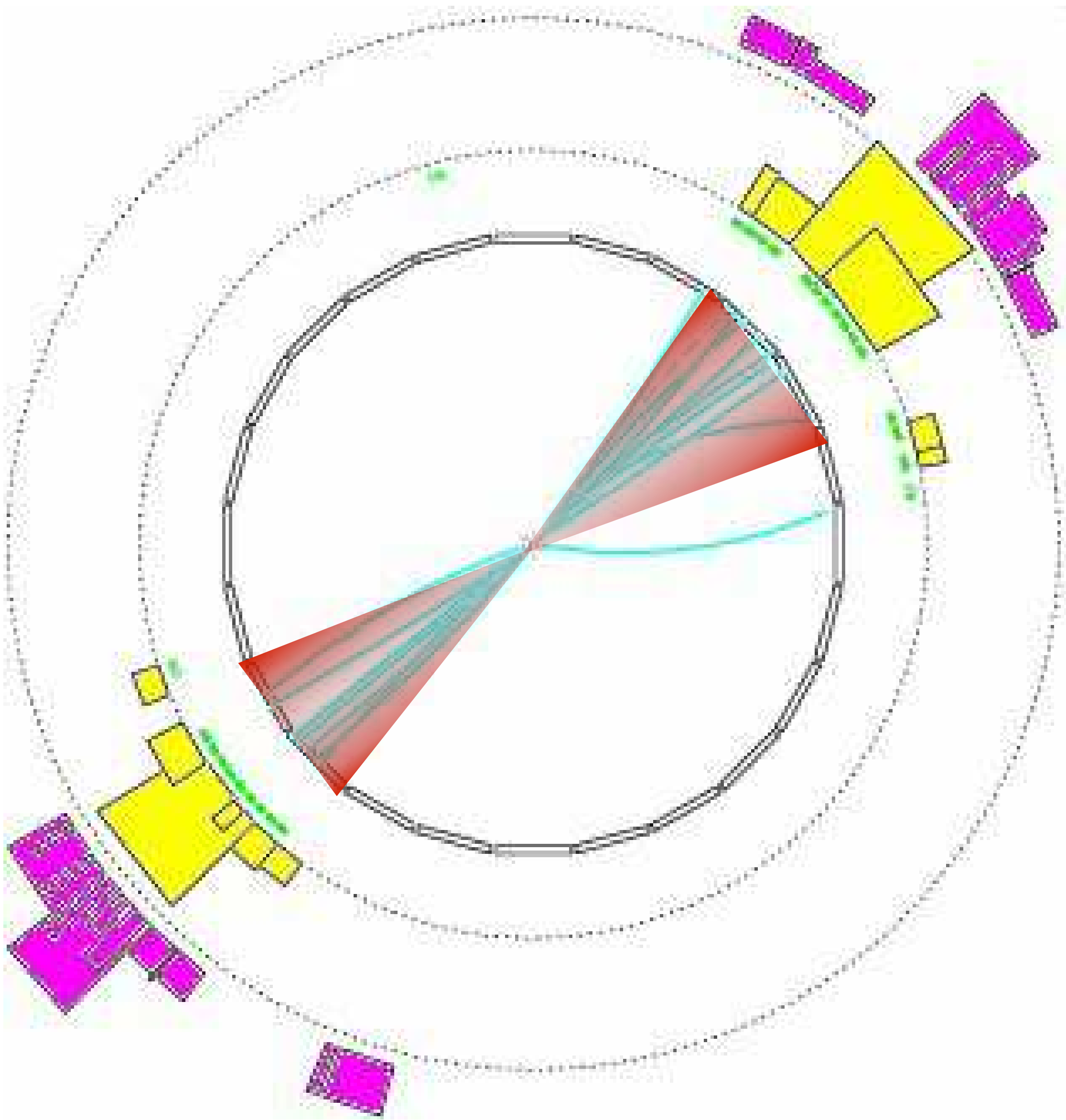
Reconstructing jets is an ambiguous task



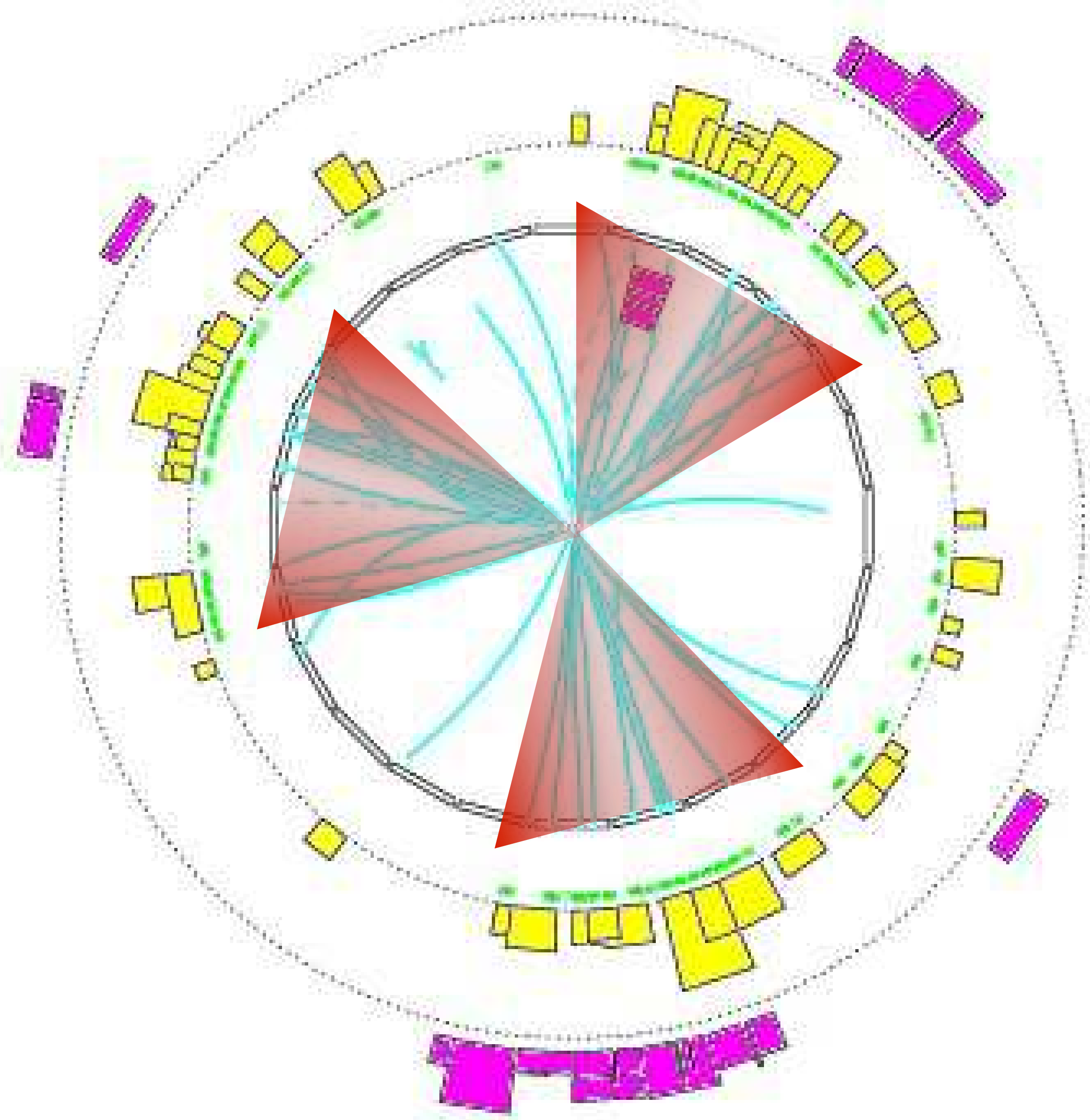


2 clear jets

Reconstructing jets is an ambiguous task

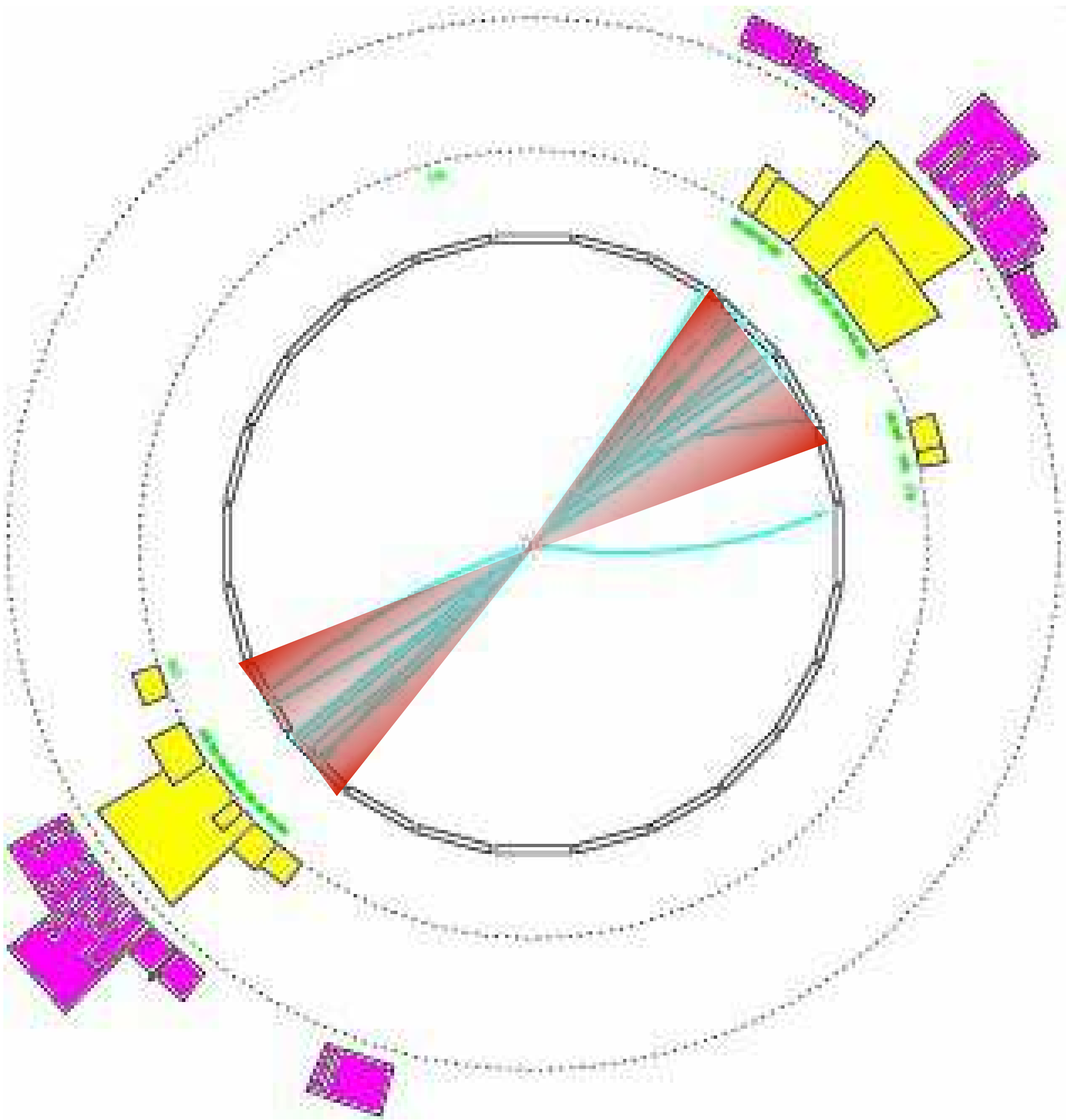


2 clear jets

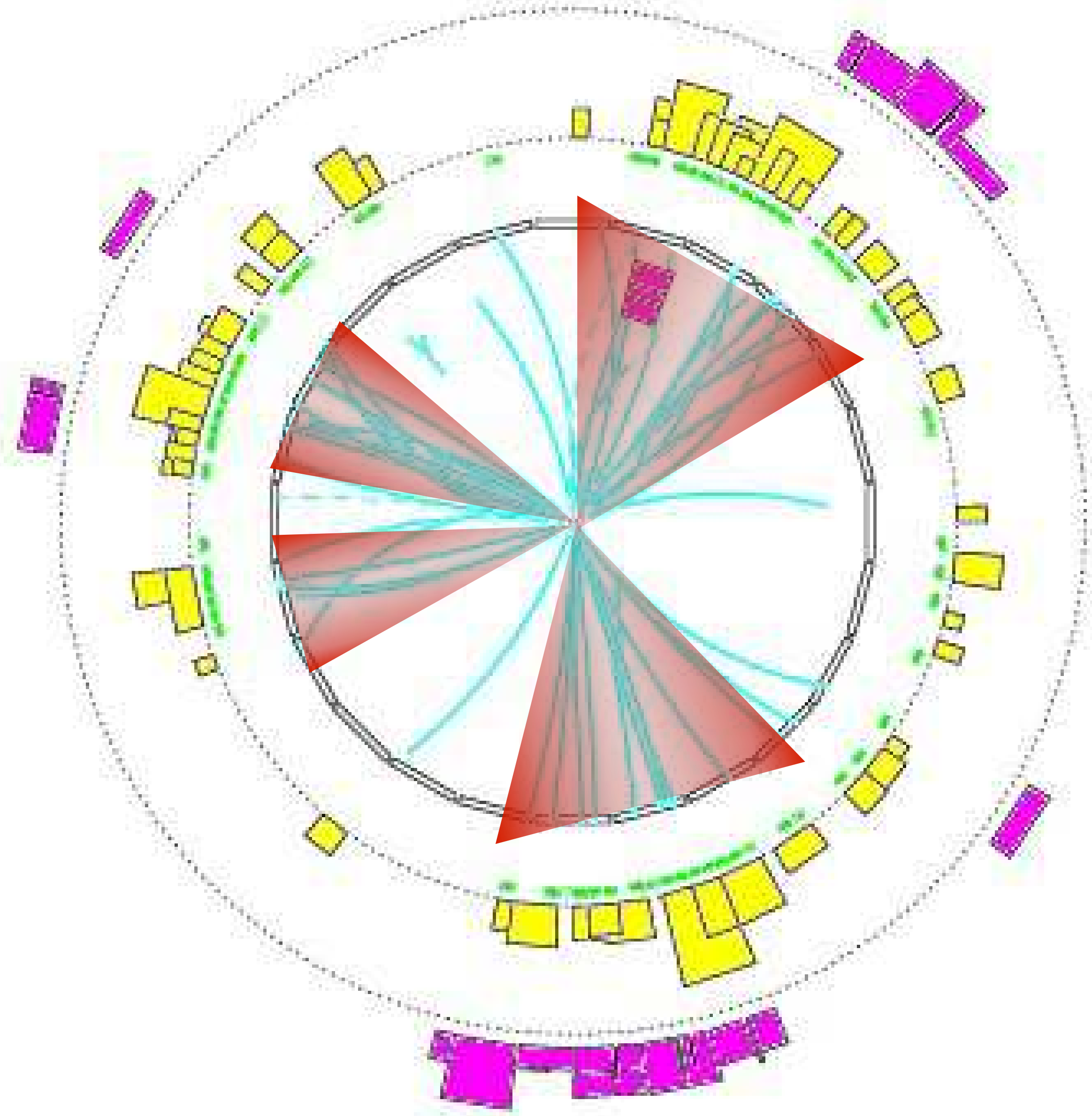


3 jets?

Reconstructing jets is an ambiguous task



2 clear jets



3 jets?
or 4 jets?

Make a choice: specify a jet definition

$$\{p_i\} \xrightarrow{\text{jet definition}} \{j_k\}$$

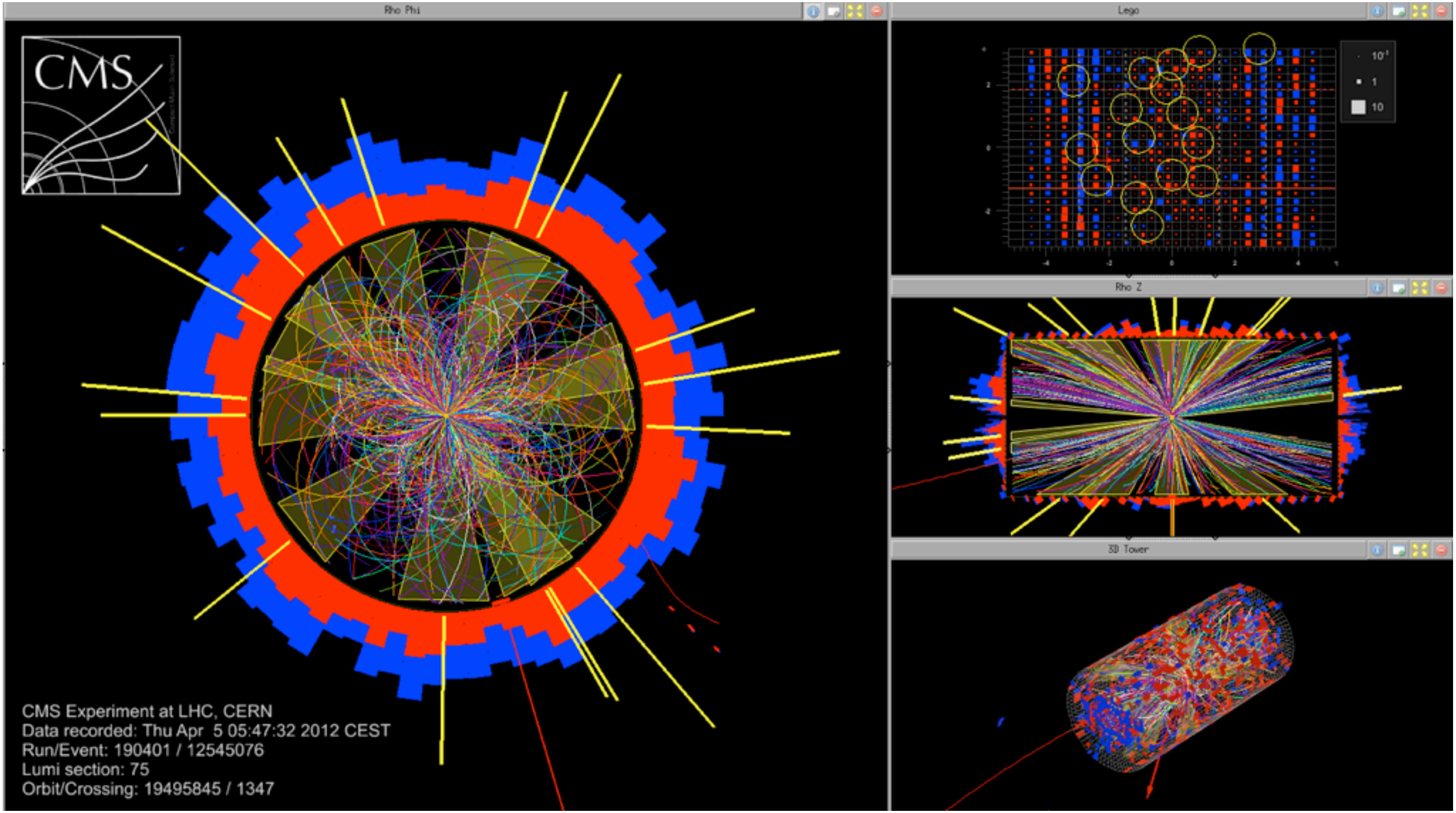
particles,
4-momenta,
calorimeter towers, ...

jets

- Which particles do you put together into a same jet?
- How do you recombine their momenta
(4-momentum sum is the obvious choice, right?)

"Jet [definitions] are legal contracts between theorists and experimentalists"
-- MJ Tannenbaum

They're also a way of organising the information in an event
1000's of particles per events, up to 20.000,000 events per second


Rho Phi
CMS
CMS Experiment at LHC, CERN
Data recorded: Thu Apr 5 05:47:32 2012 CEST
Run/Event: 190401 / 12545076
Lumi section: 75
Orbit/Crossing: 19495845 / 1347
Lego
Rho Z
3D Tower

Cluster analysis

From Wikipedia, the free encyclopedia

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

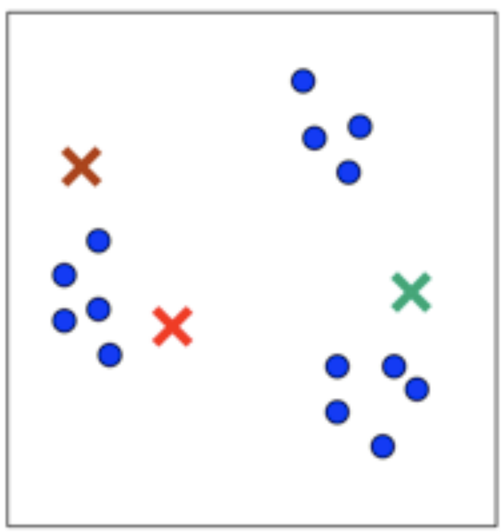
Contents [hide]

There is no objectively "correct" clustering algorithm, but […] "clustering is in the eye of the beholder."[1] The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally, unless there is a mathematical reason to prefer one cluster model over another.

Partitioning / centroid-based clustering [cone algorithms]

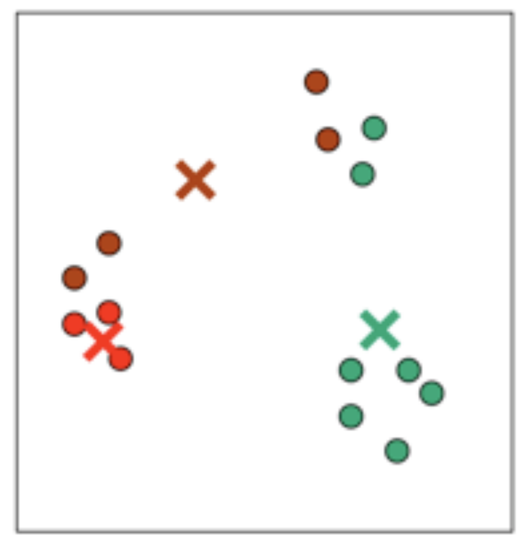
Example of a partitional algorithm

1) Choose K centroids at random
2) Assign objects to closest centroid, forming K clusters
3) Calculate centroid (mean of distances) of each cluster, update centroids
4) Check if an object in a cluster is closer to another centroid. Reallocate in case.
5) Repeat from step 3 until no object changes cluster anymore.

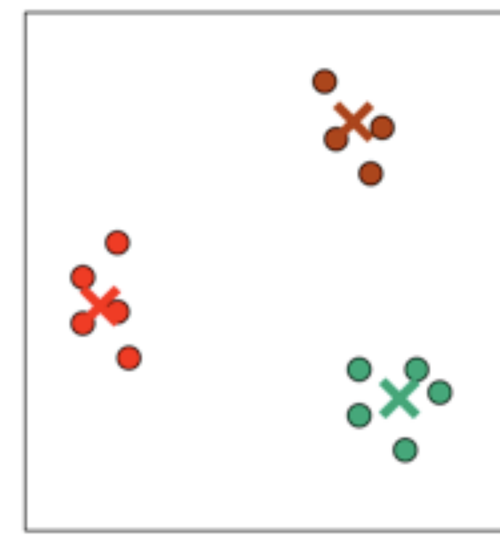


Step 1
(random centroids)

Step 2
(allocate objects)



Step 3
(move centroids)

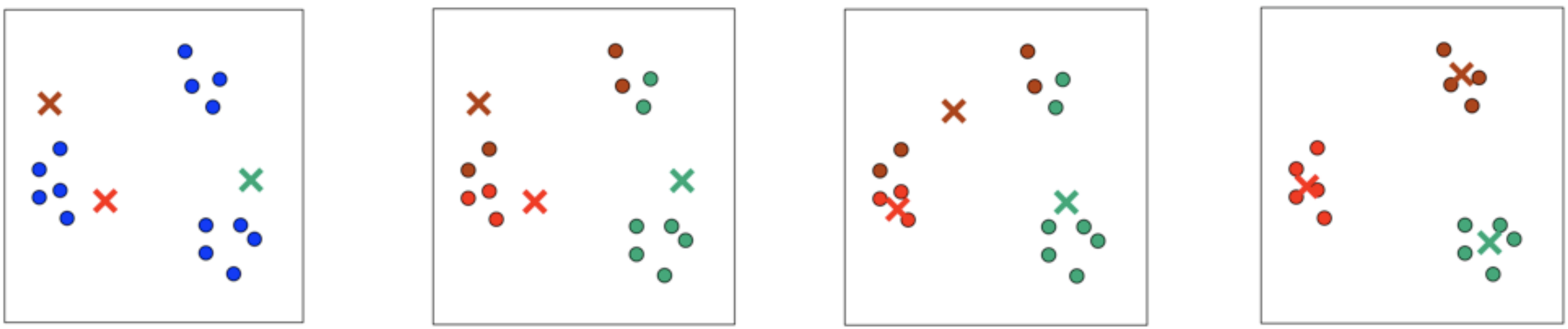


Step 5
(end of iteration)

[Figures by E. Garrett-Mayer]

Example of a partitional algorithm

1) Choose K centroids at random
2) Assign objects to closest centroid, forming K clusters
3) Calculate centroid (mean of distances) of each cluster, update centroids
4) Check if an object in a cluster is closer to another centroid. Reallocate in case.
5) Repeat from step 3 until no object changes cluster anymore.



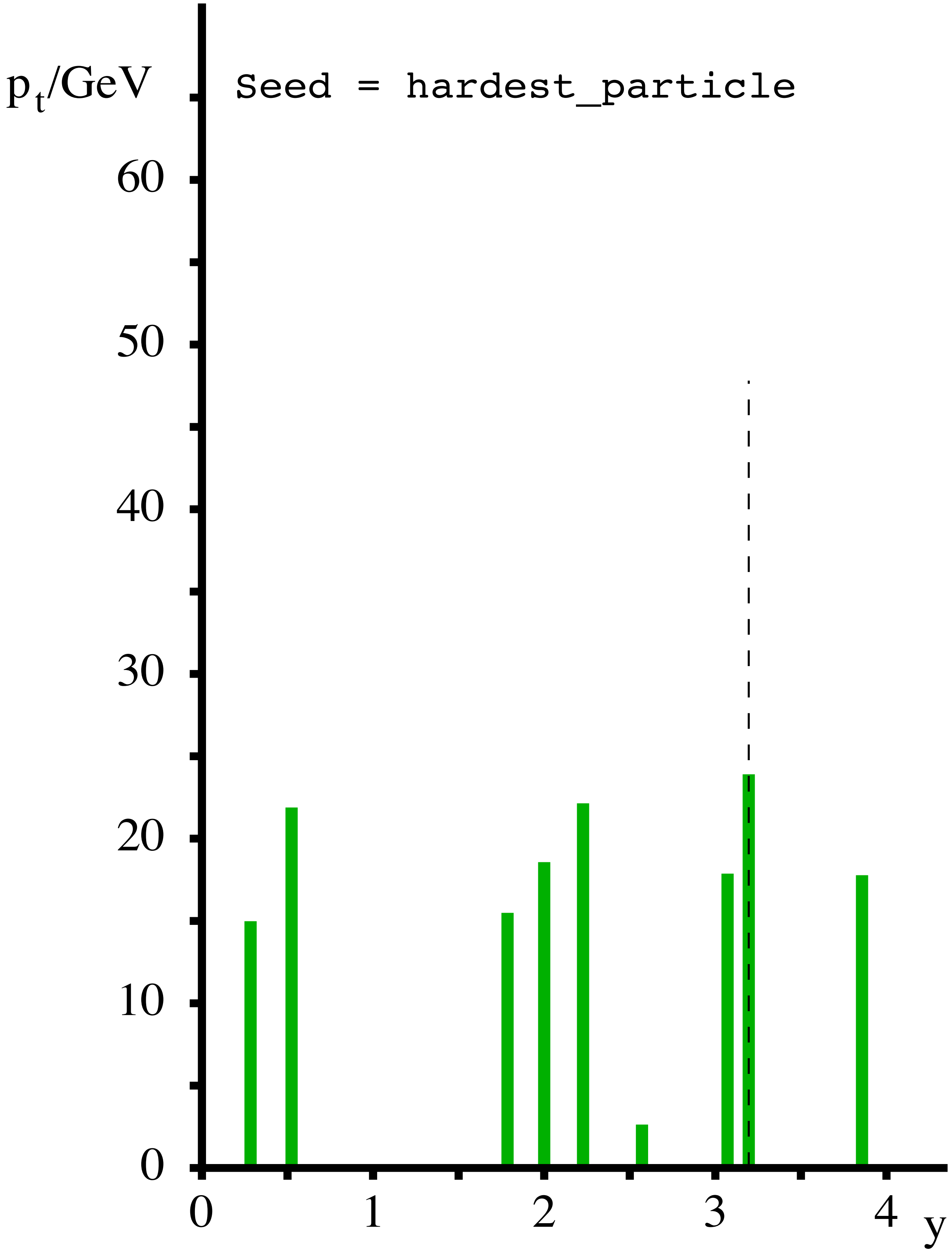
Step 1 (random centroids) | Step 2 (allocate objects) | Step 3 (move centroids) | Step 5 (end of iteration)

[Figures by E. Garrett-Mayer]

One of the main shortcomings:

result of final convergence can be highly sensitive to choice of initial seeds.

Also, the concept of 'mean distance' (to calculate the centroid) must be defined.



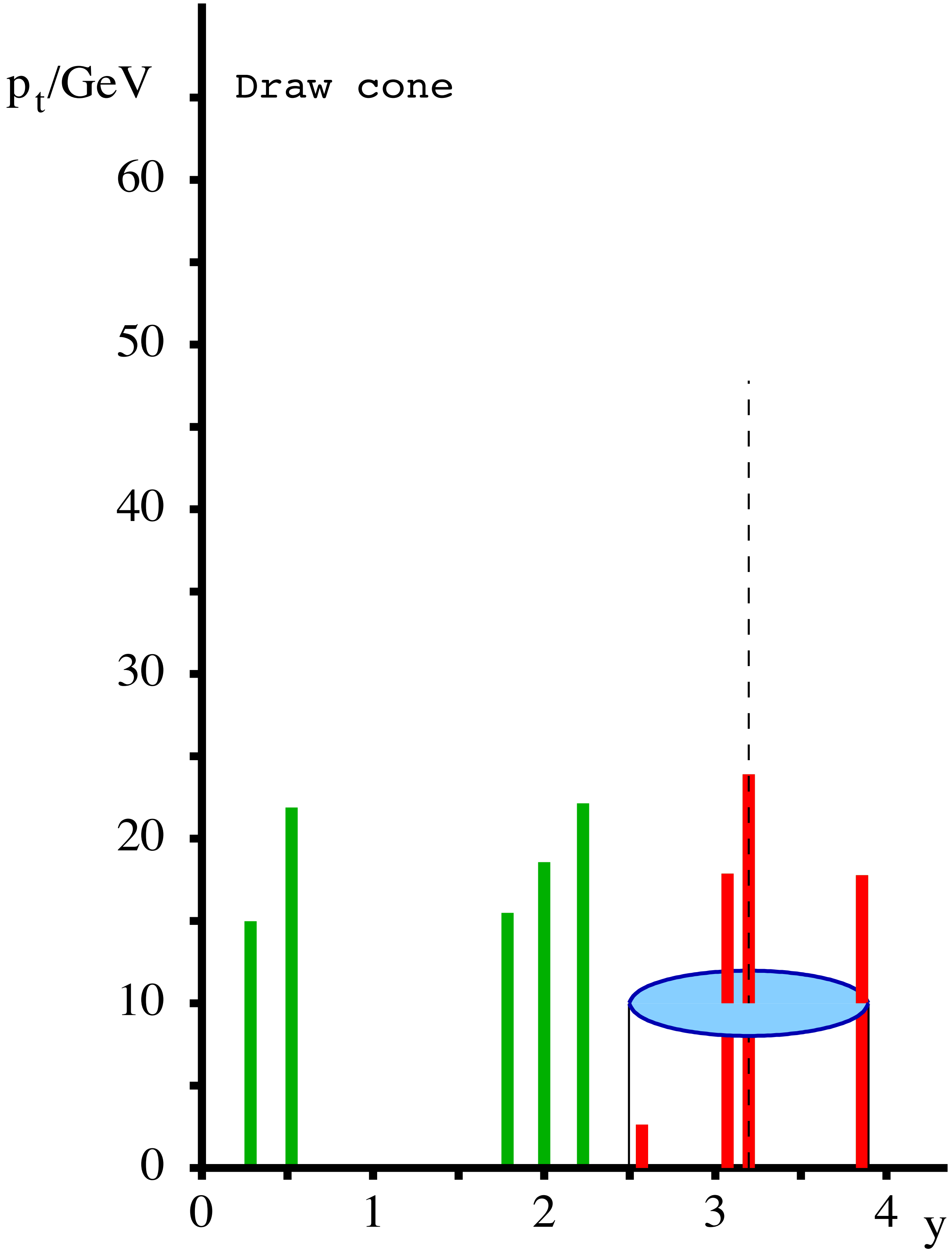
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

- "Hardest particle" is collinear unsafe more right away...



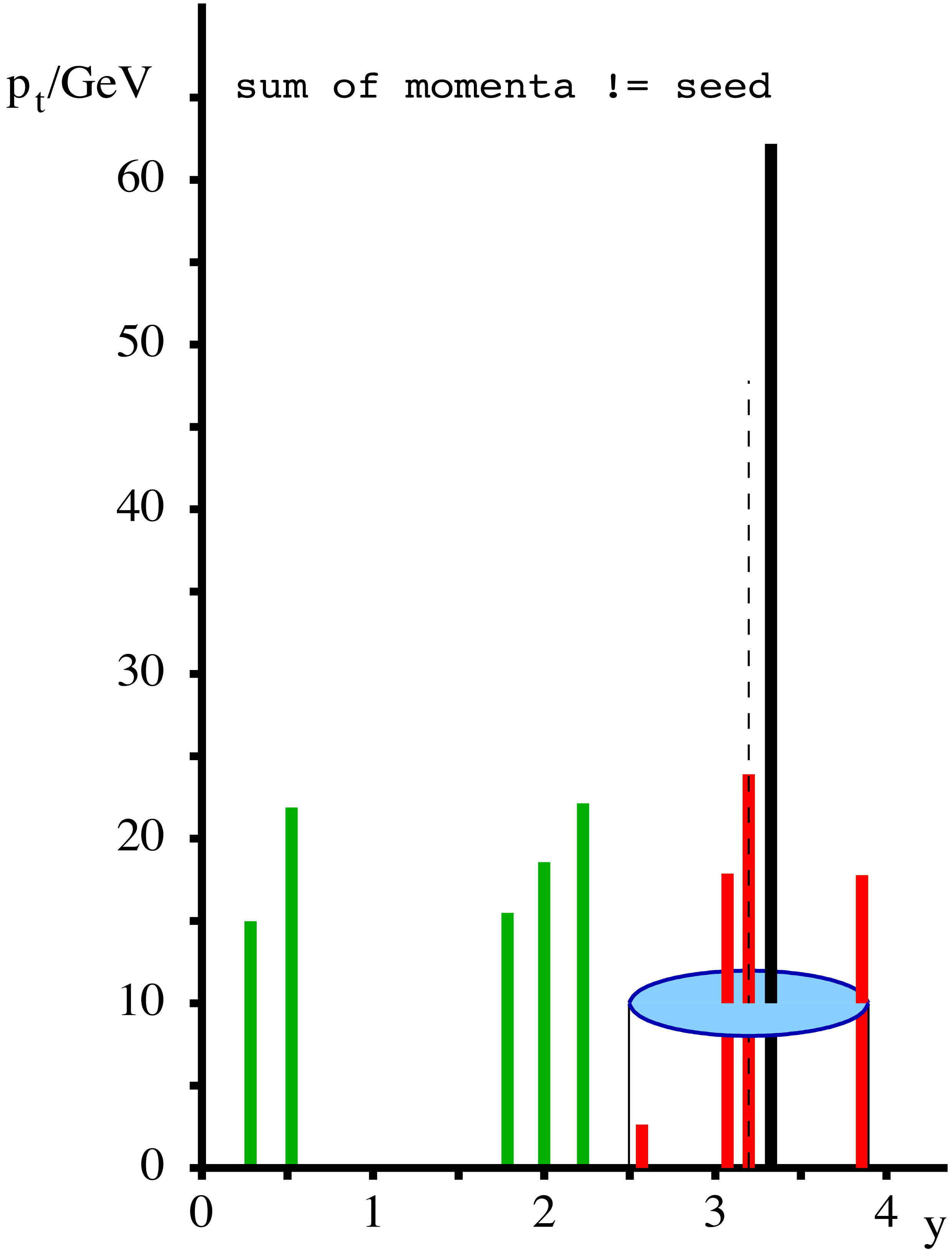
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

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



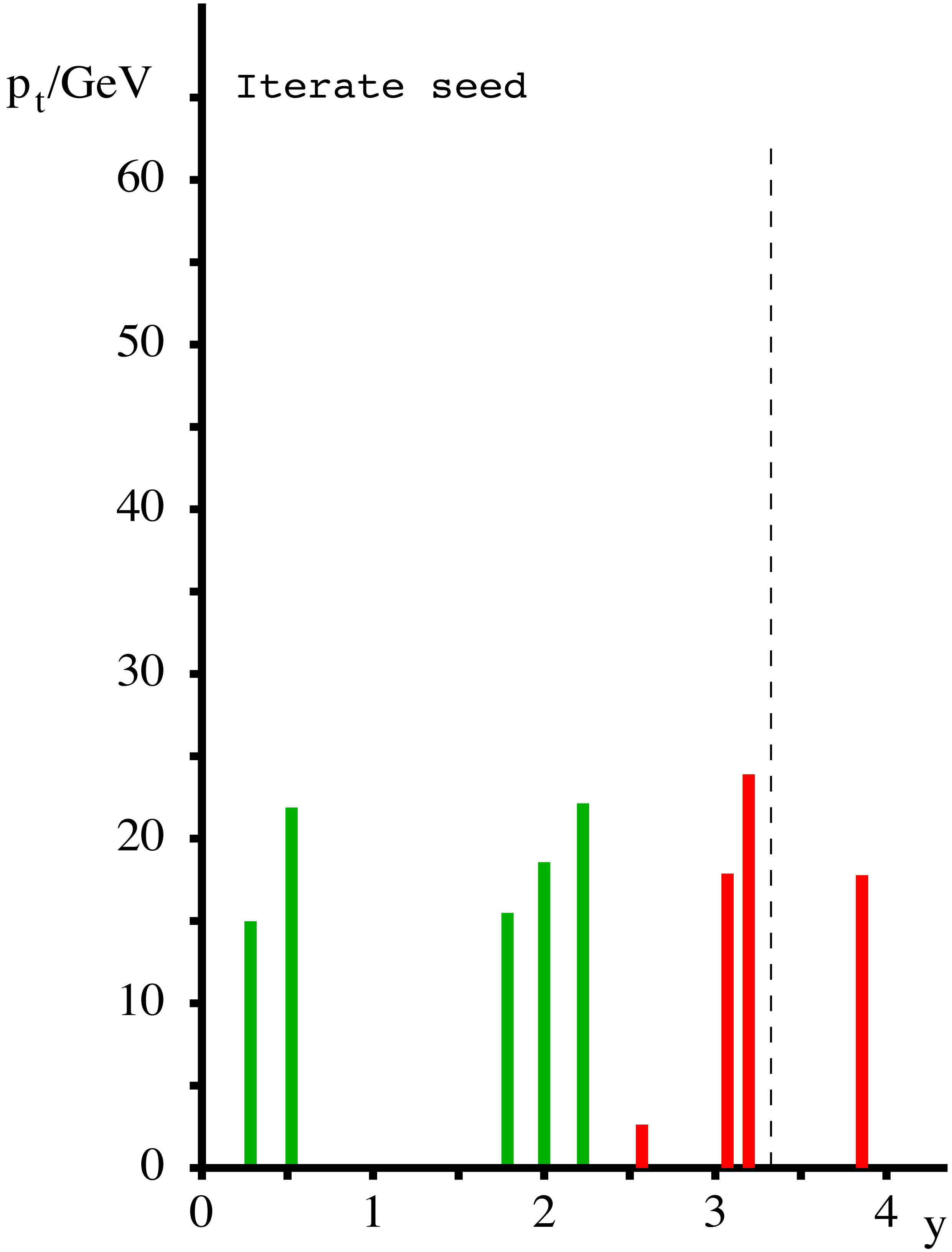
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

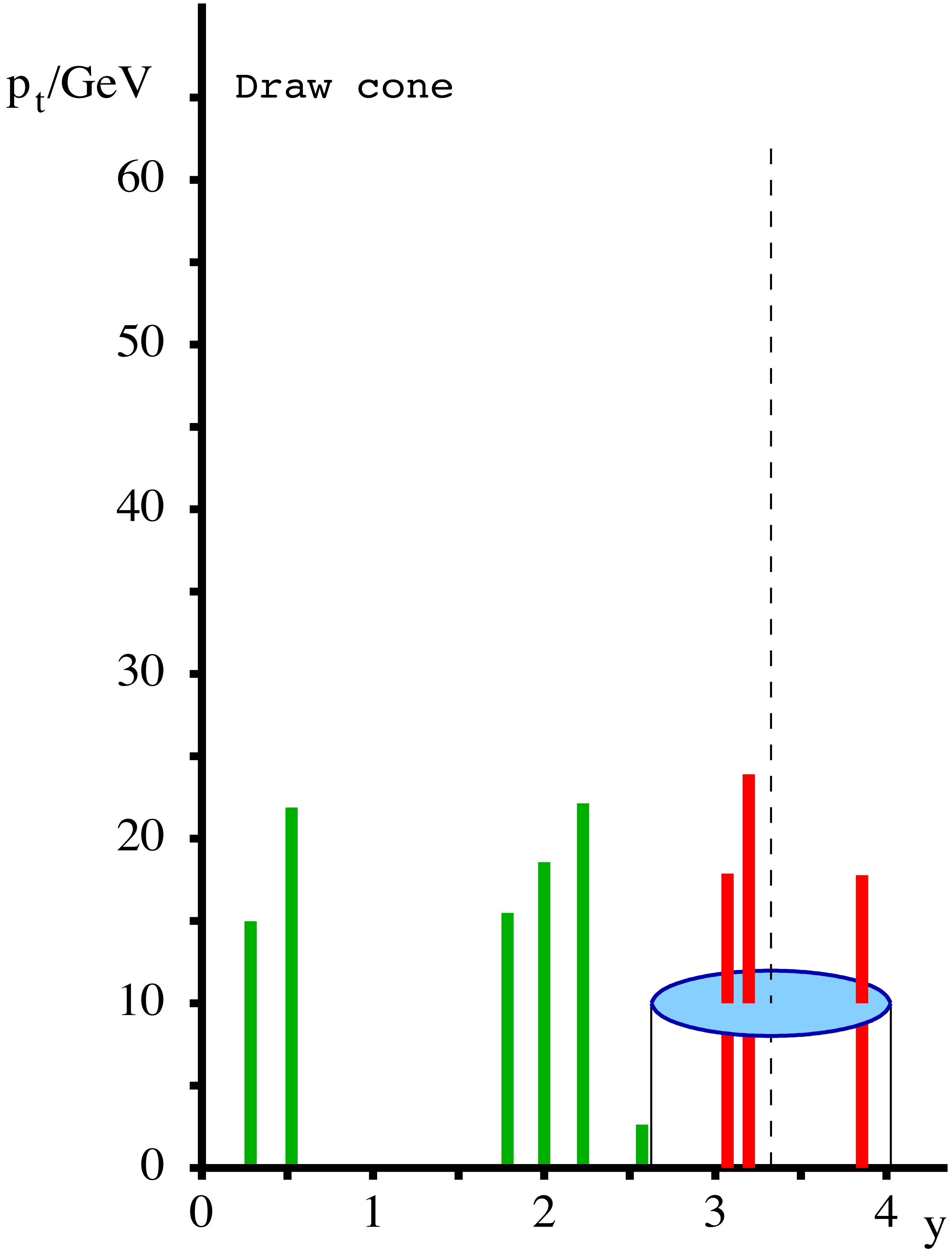
- "Hardest particle" is collinear unsafe more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

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

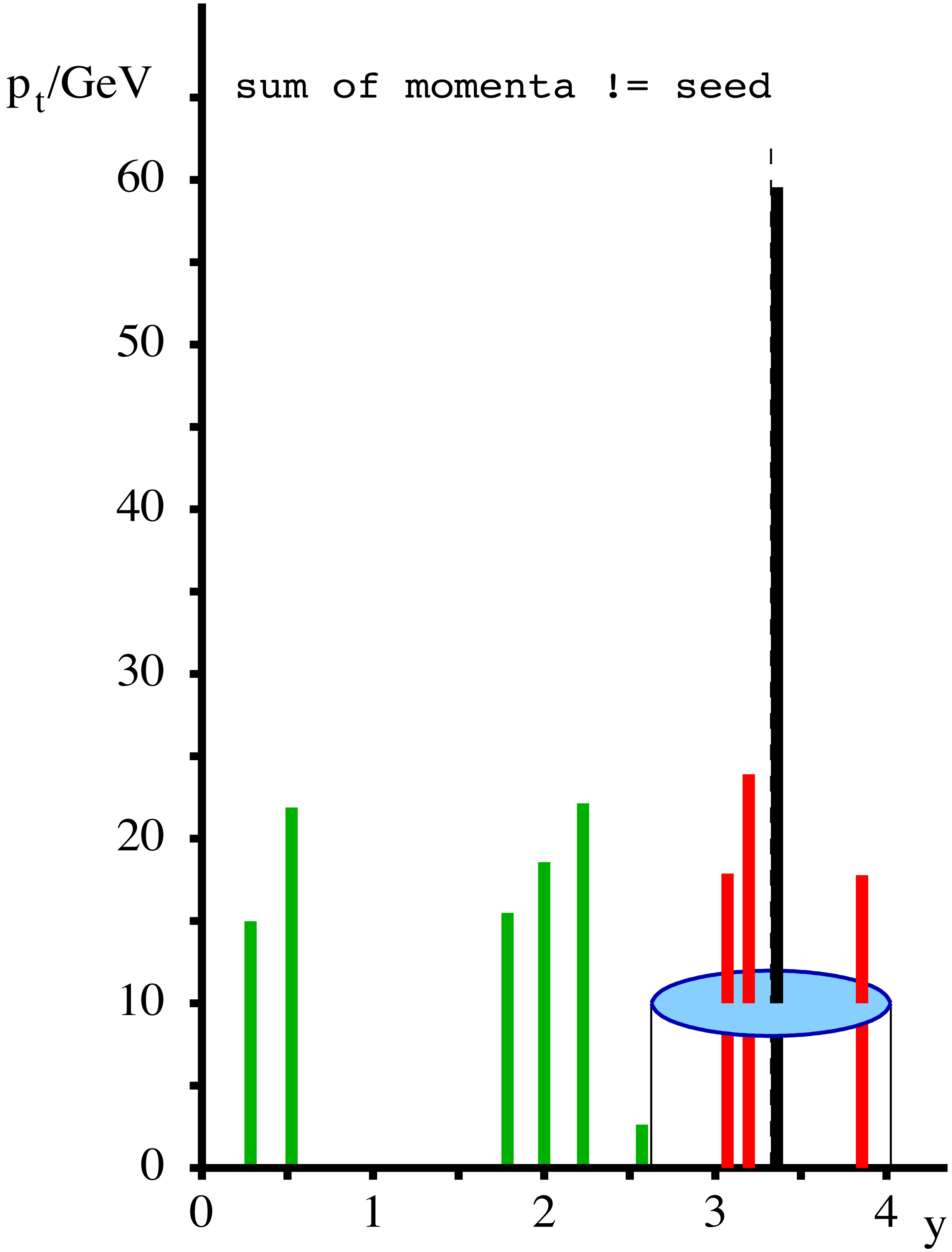


One of the simplest of the cone algs

e.g. CMS iterative cone

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

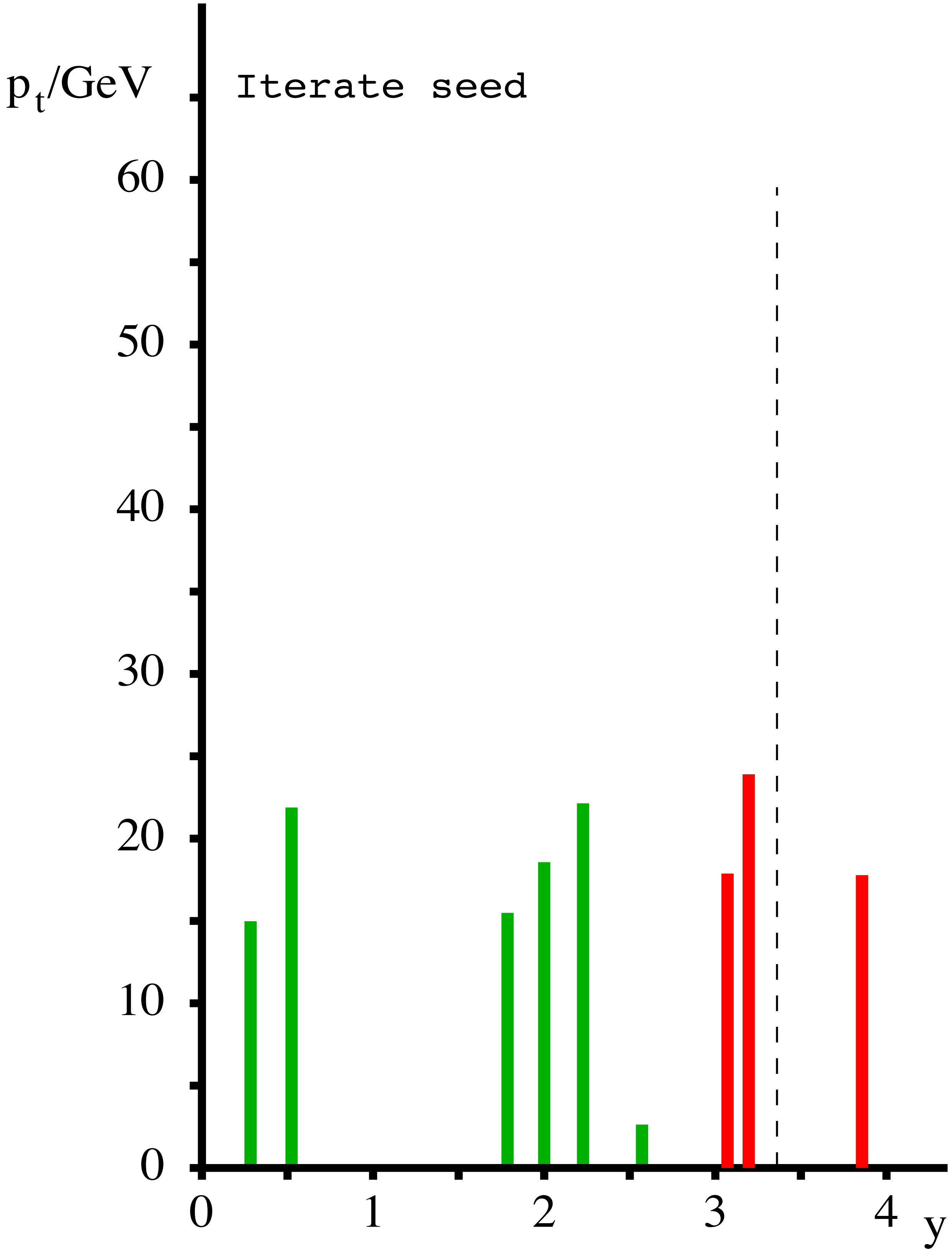
Iterative Cone, Prog Removal (IC-PR)



One of the simplest of the cone algs

e.g. CMS iterative cone

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



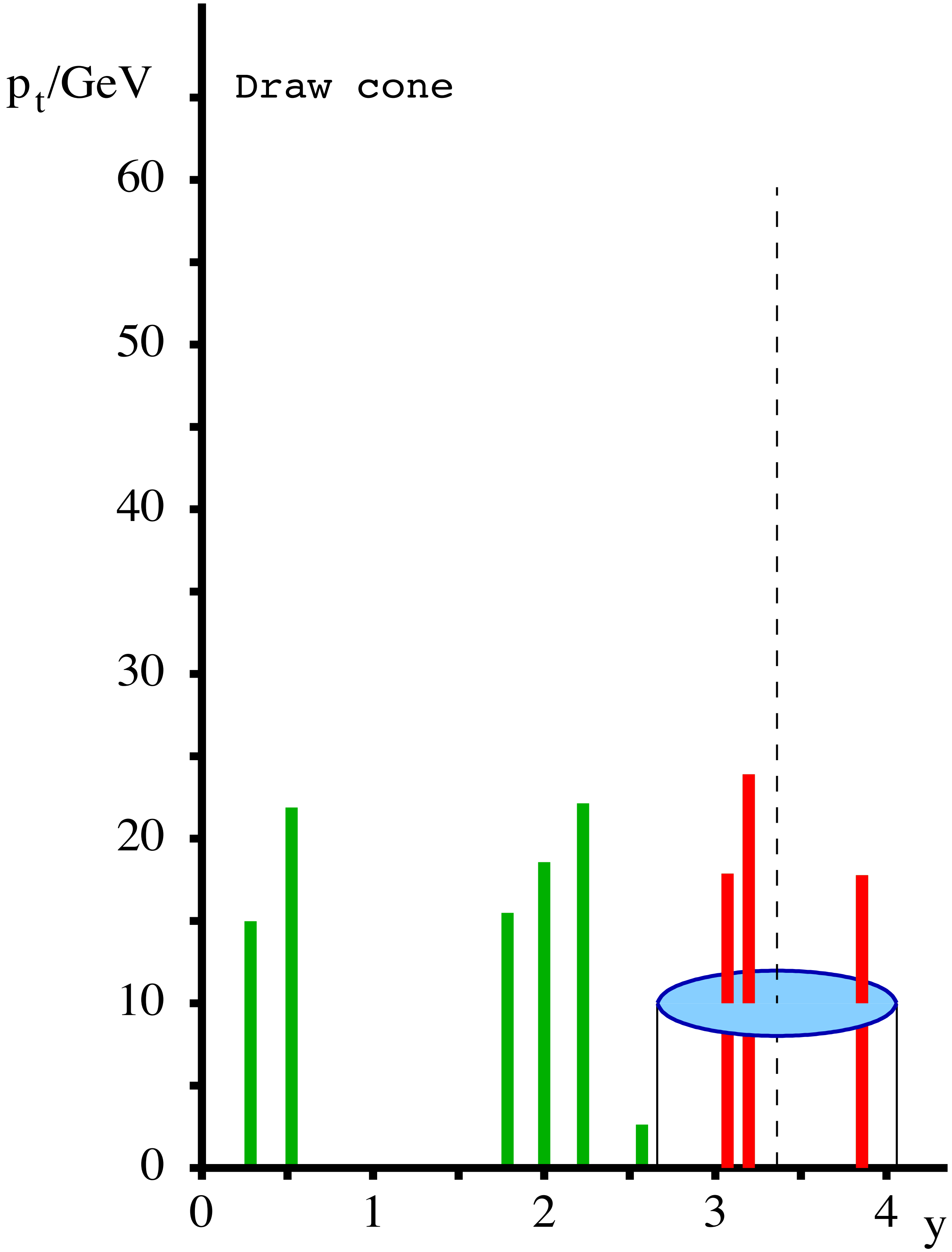
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

- "Hardest particle" is collinear unsafe more right away...



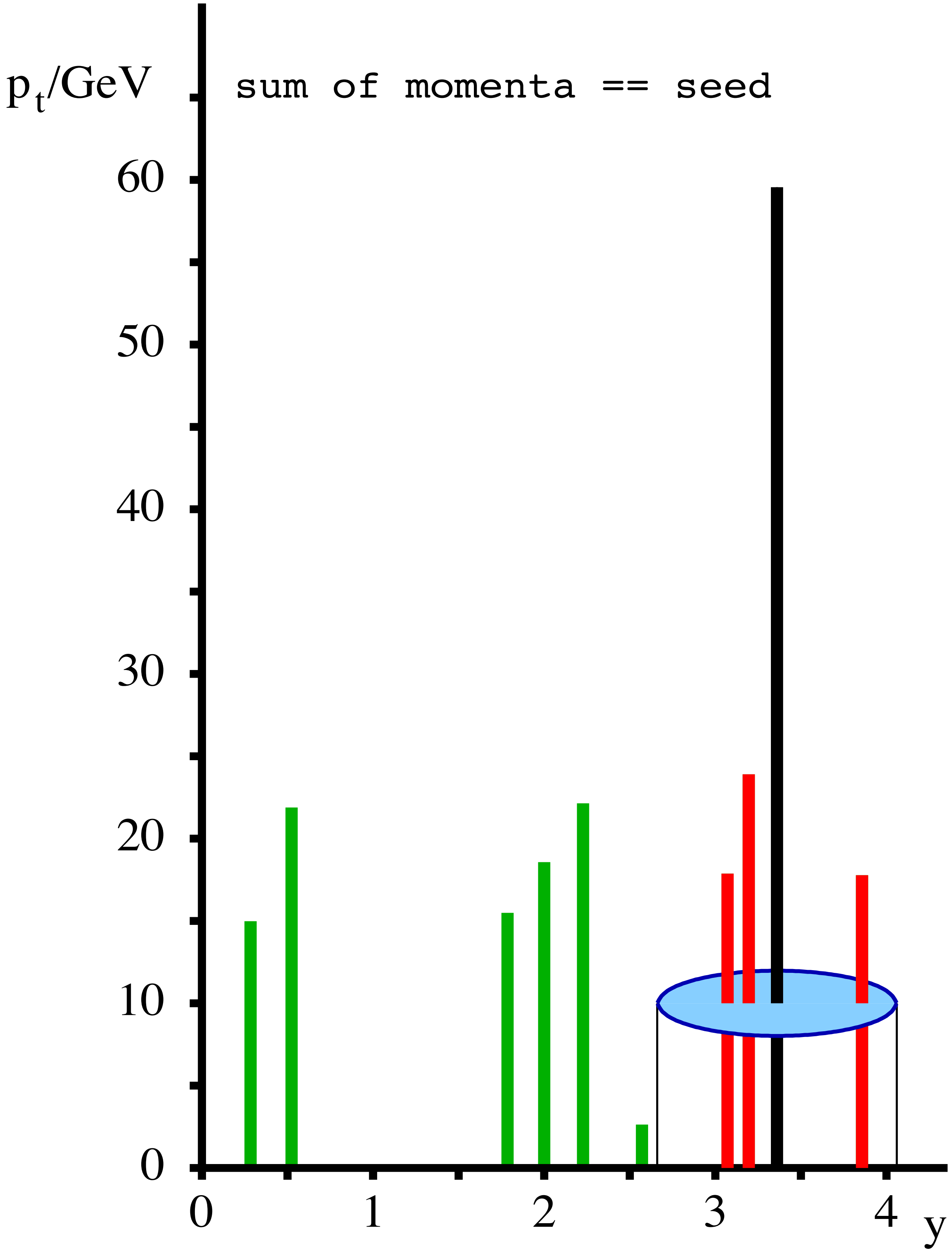
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

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



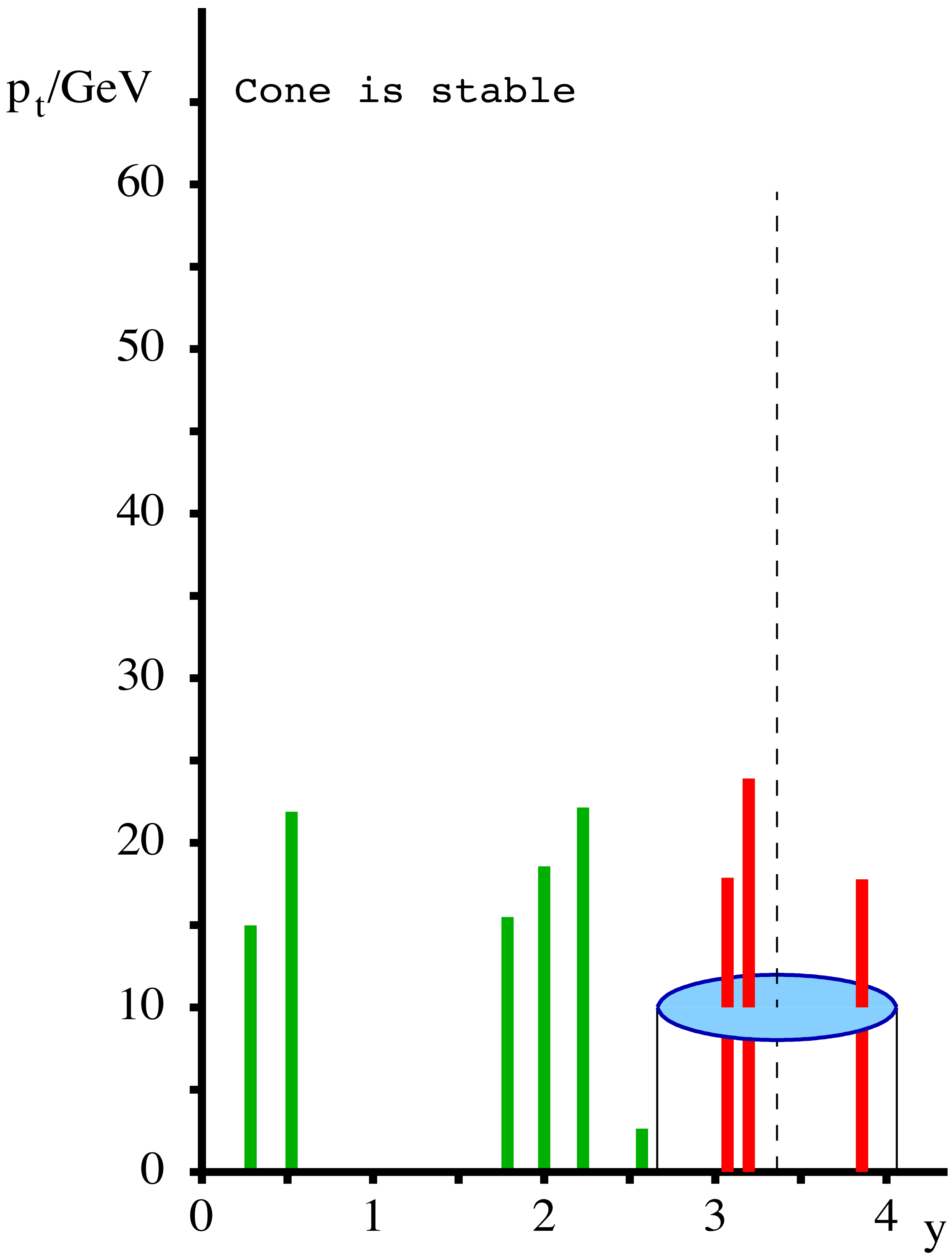
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

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



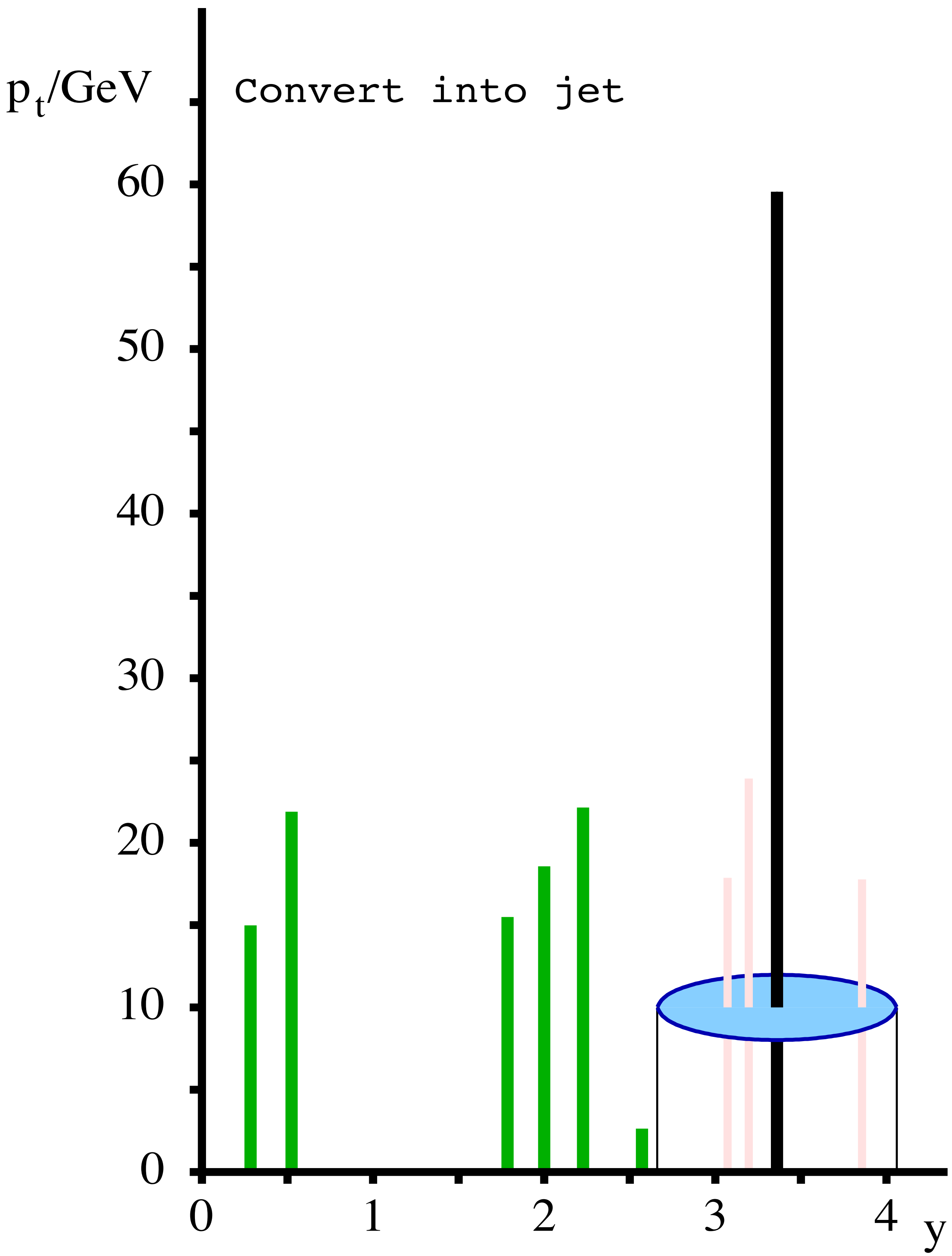
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

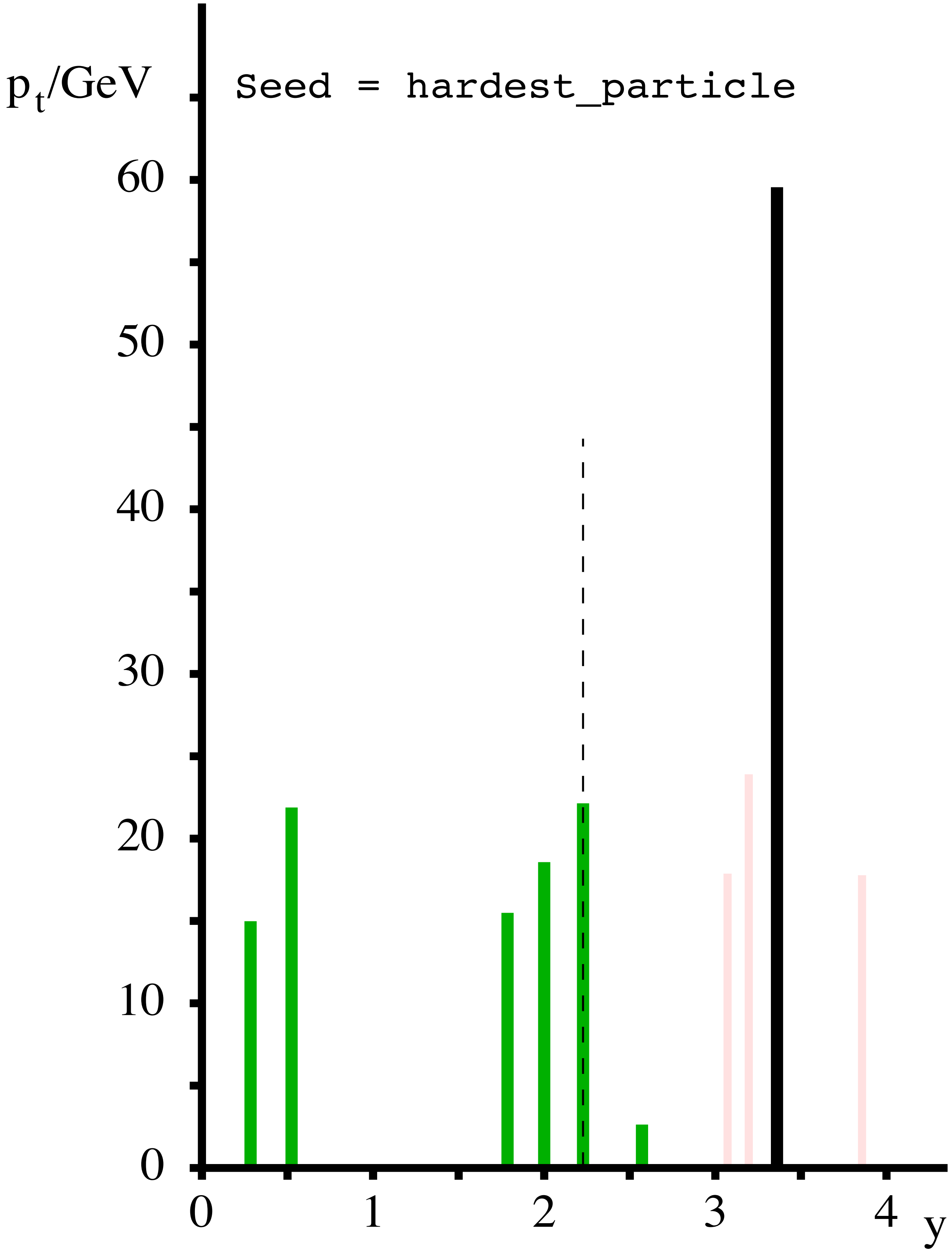
- "Hardest particle" is collinear unsafe more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

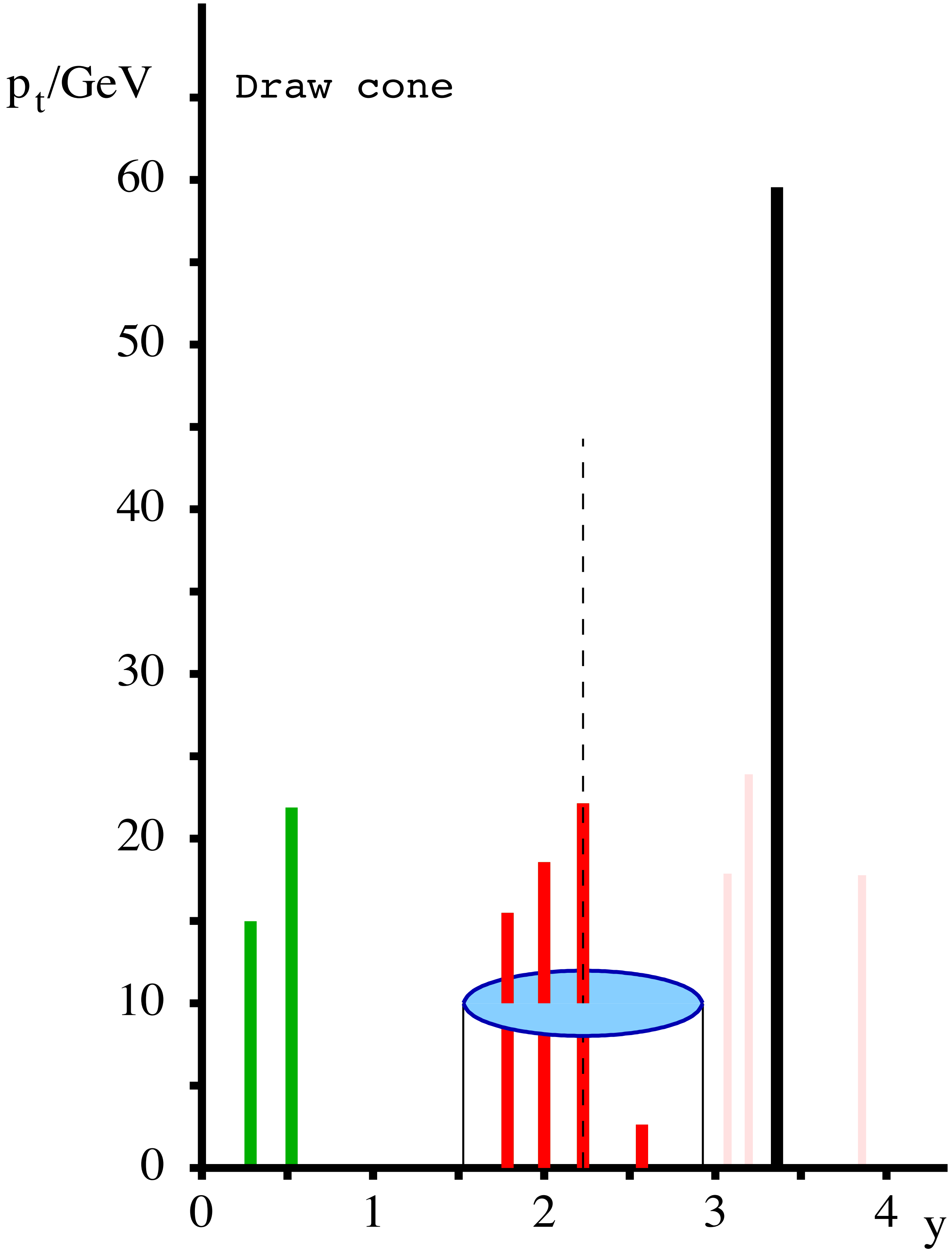
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

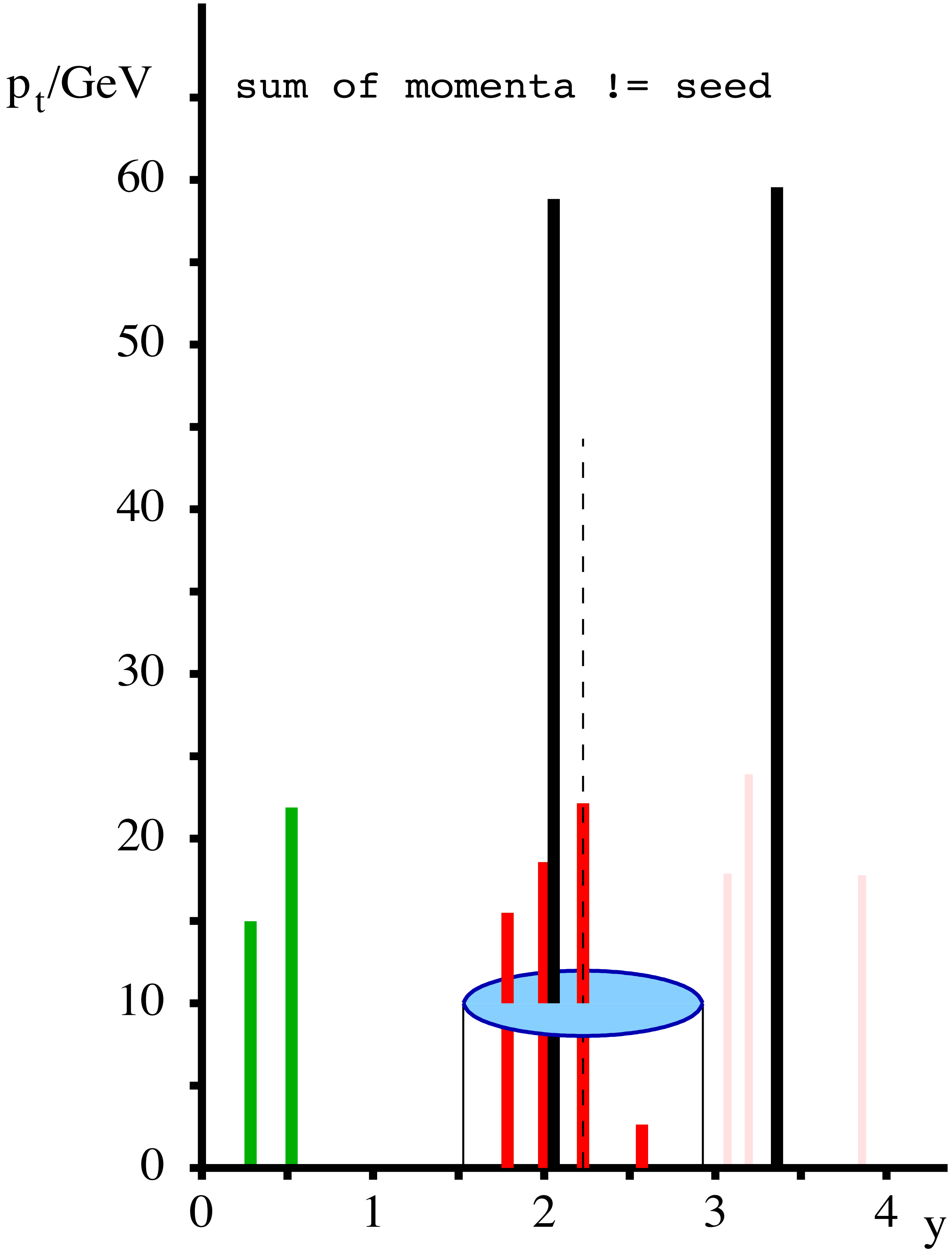
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

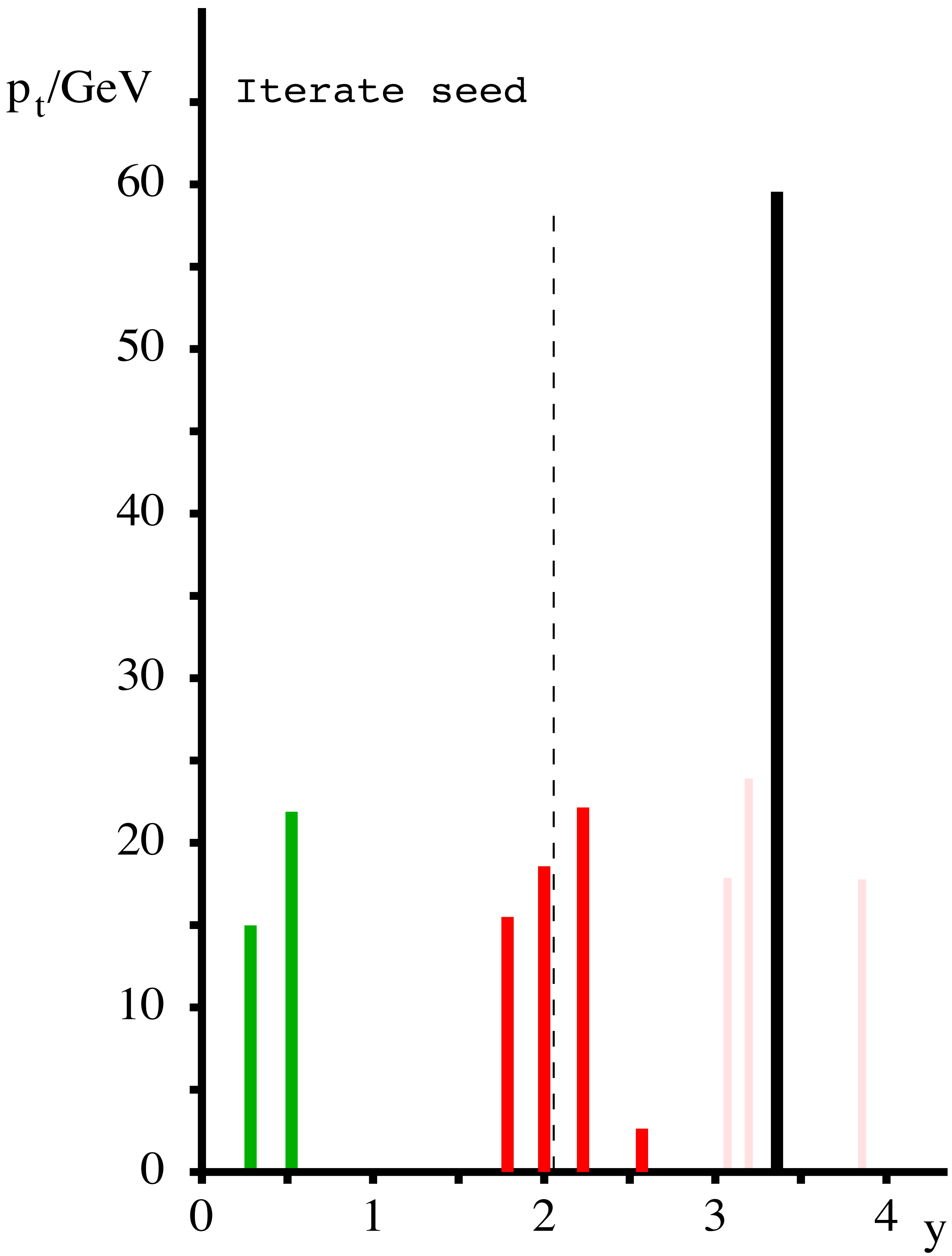
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

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



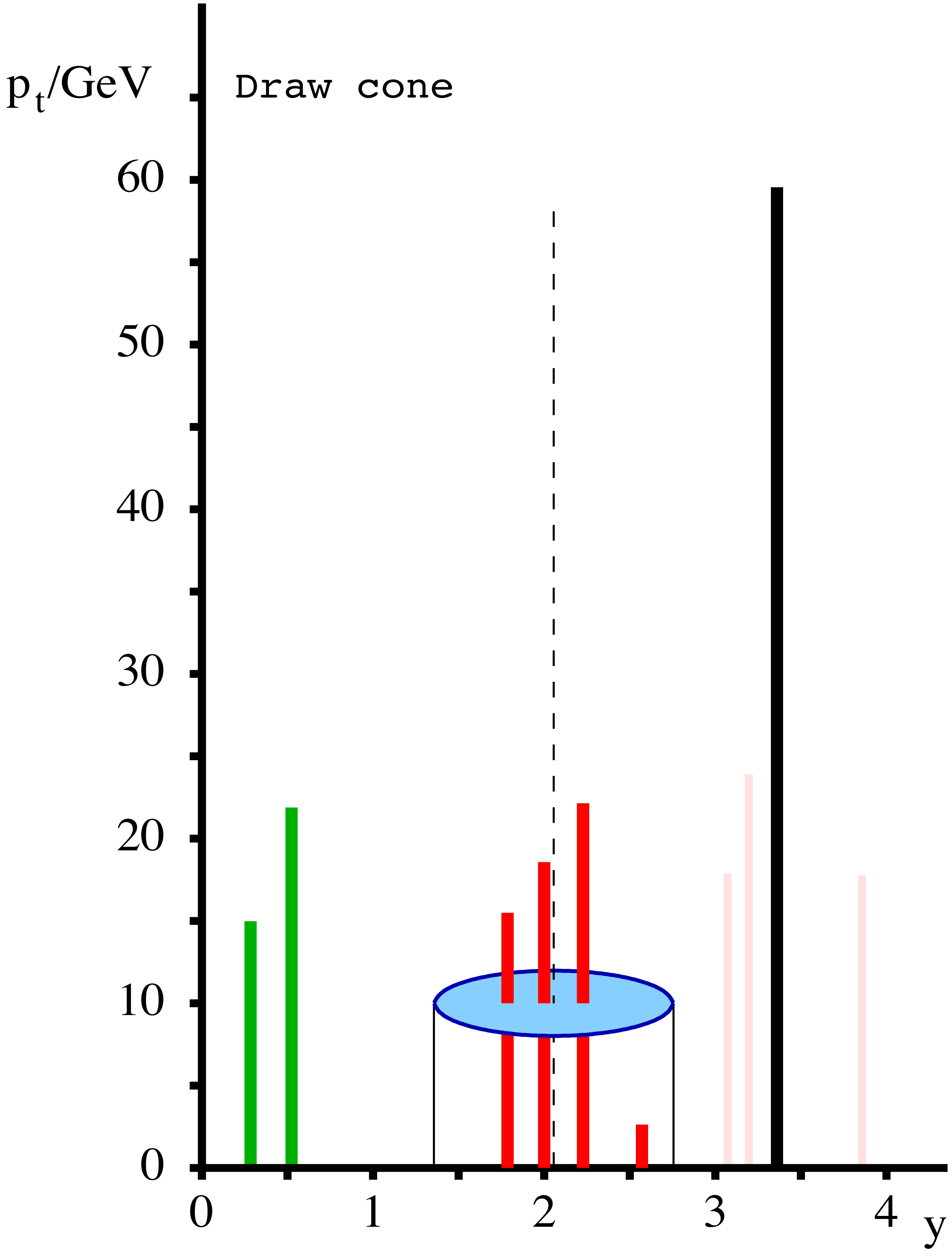
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

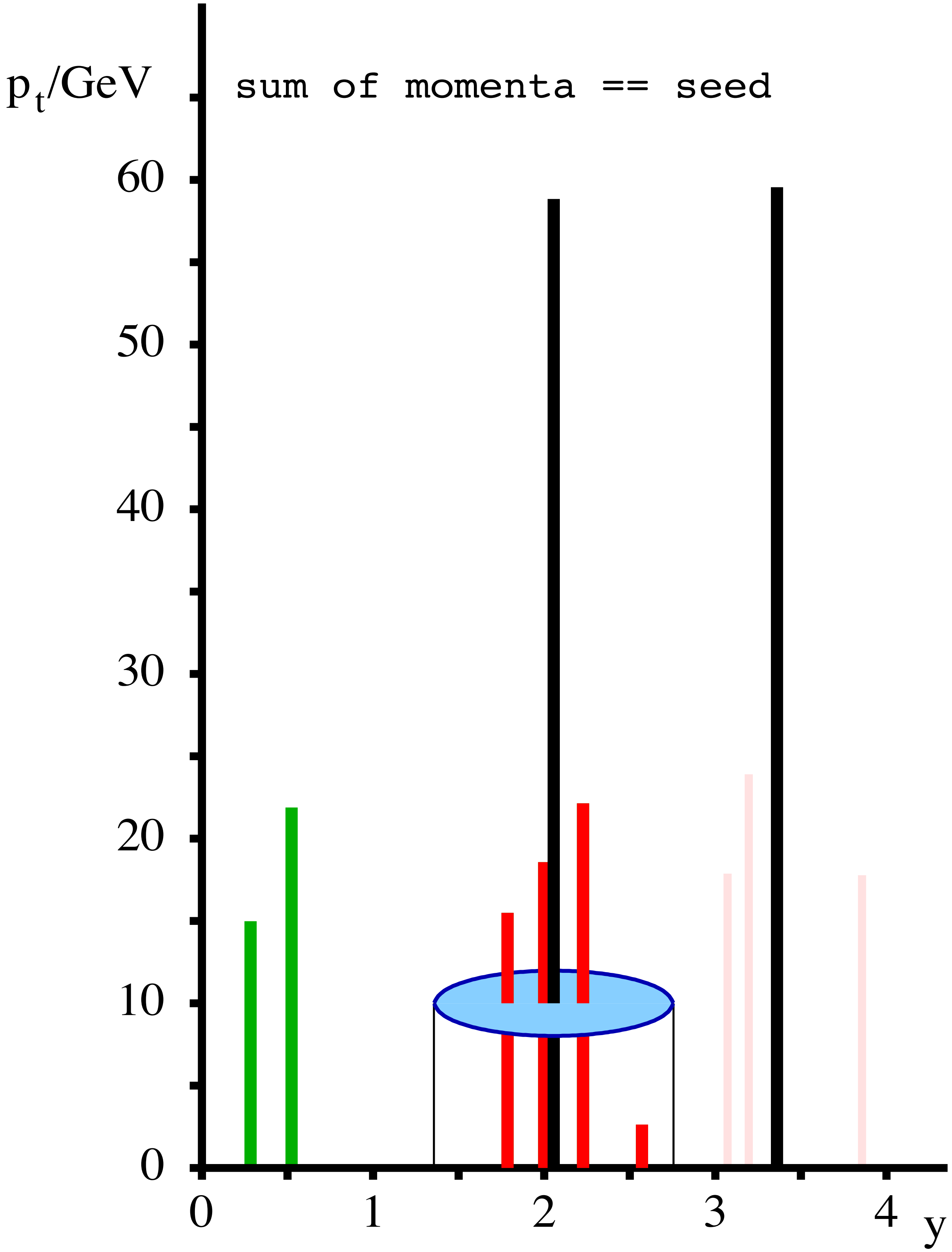
- “Hardest particle” is collinear unsafe more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

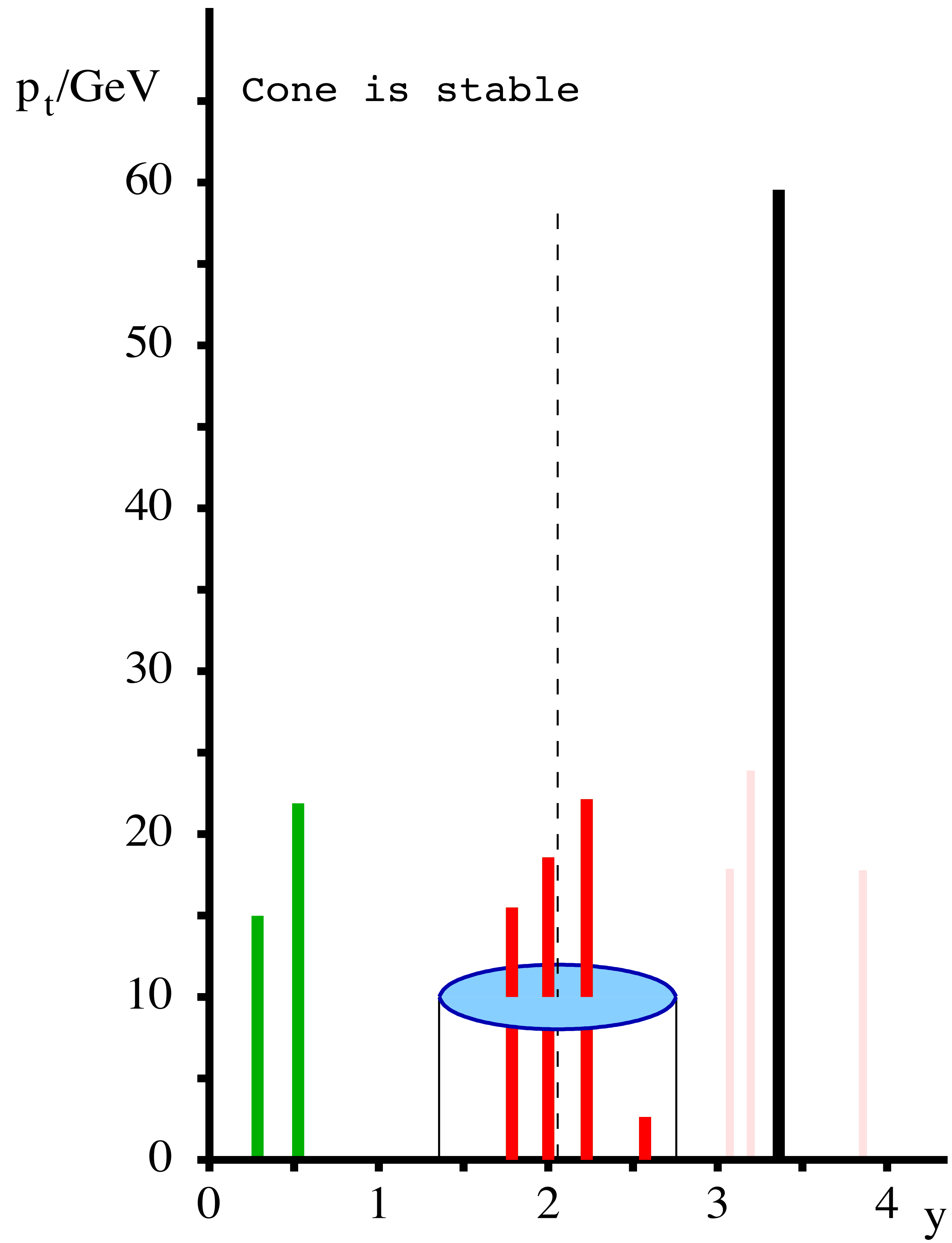
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

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

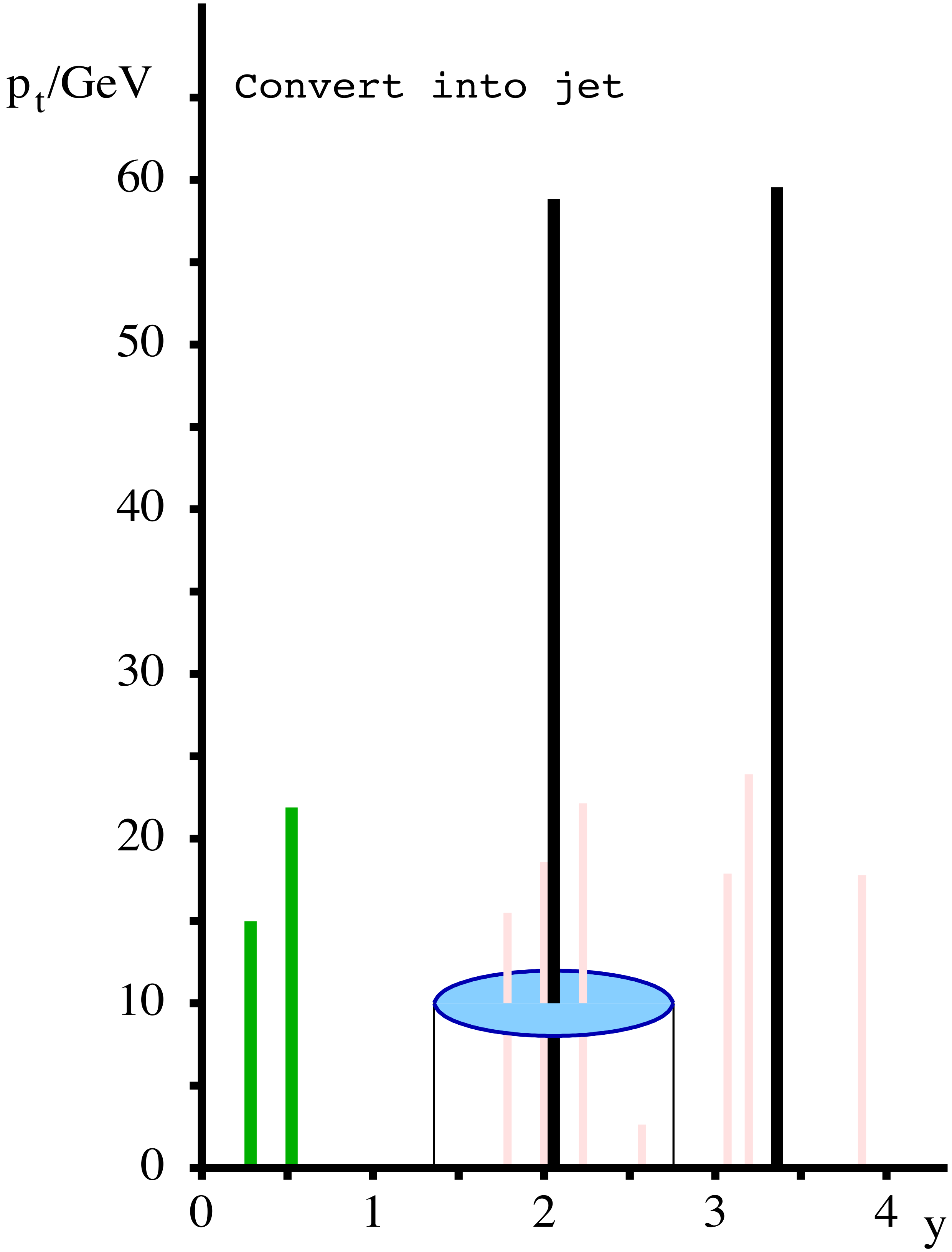


One of the simplest of the cone algs

e.g. CMS iterative cone

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

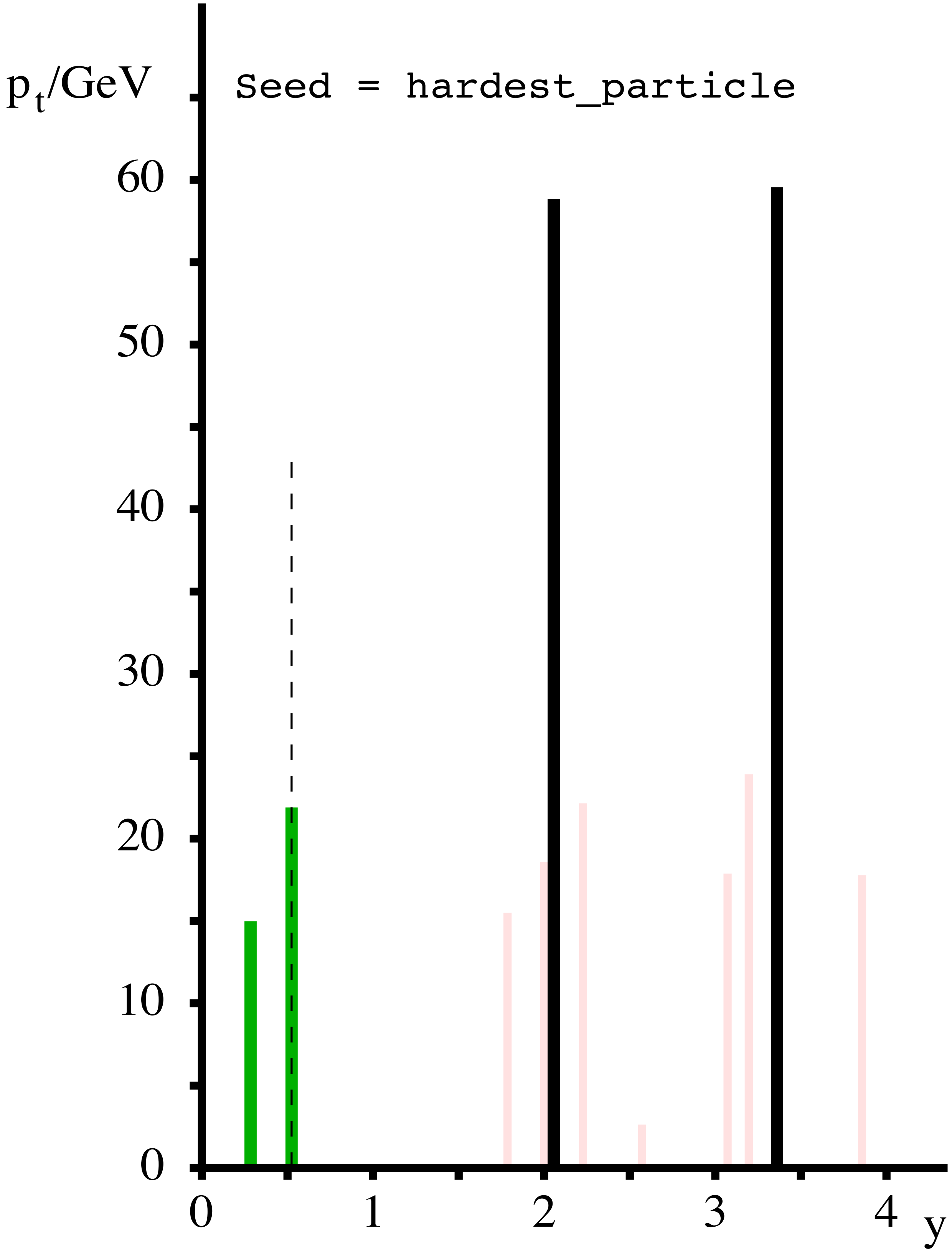
Iterative Cone, Prog Removal (IC-PR)



One of the simplest of the cone algs

e.g. CMS iterative cone

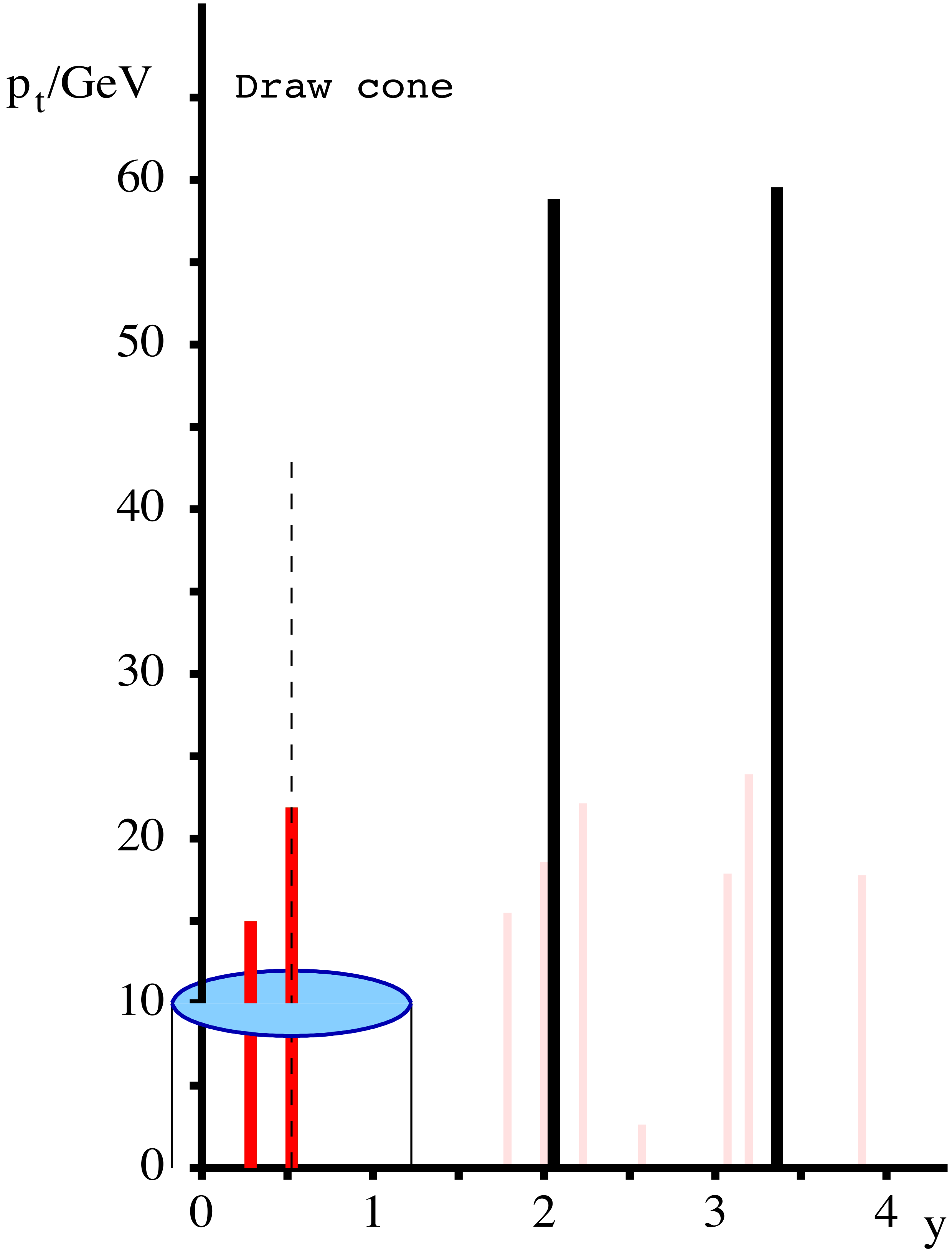
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

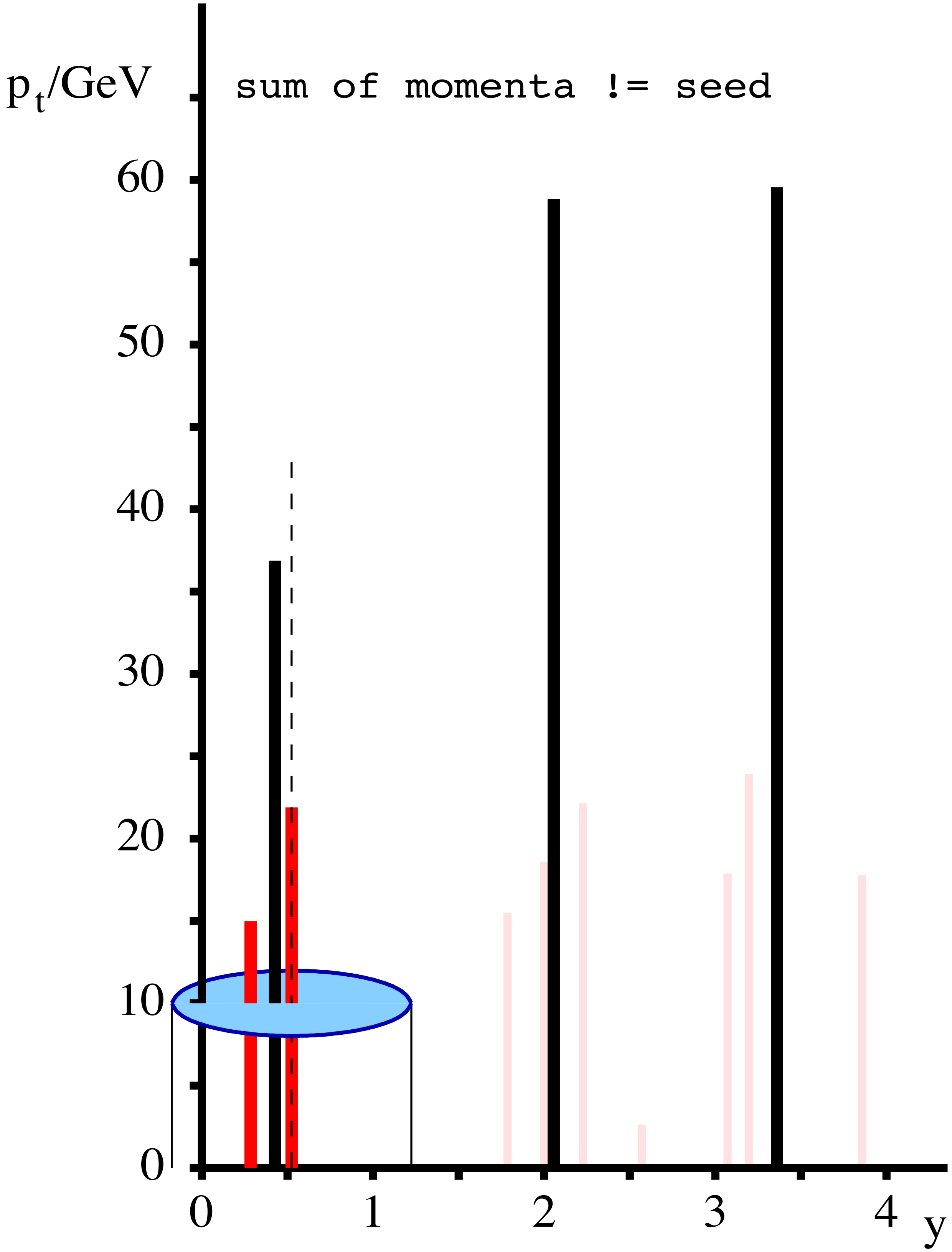
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

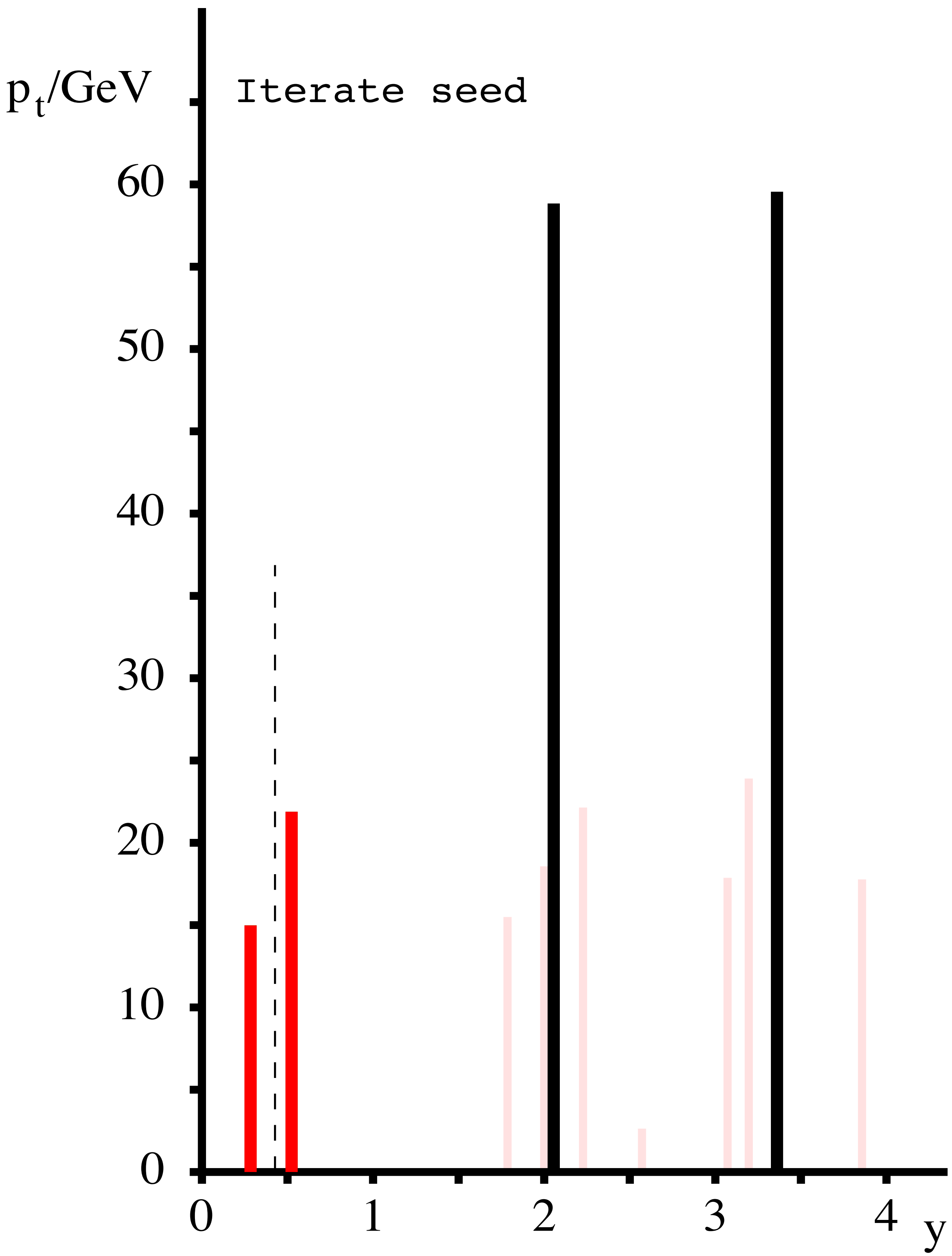
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

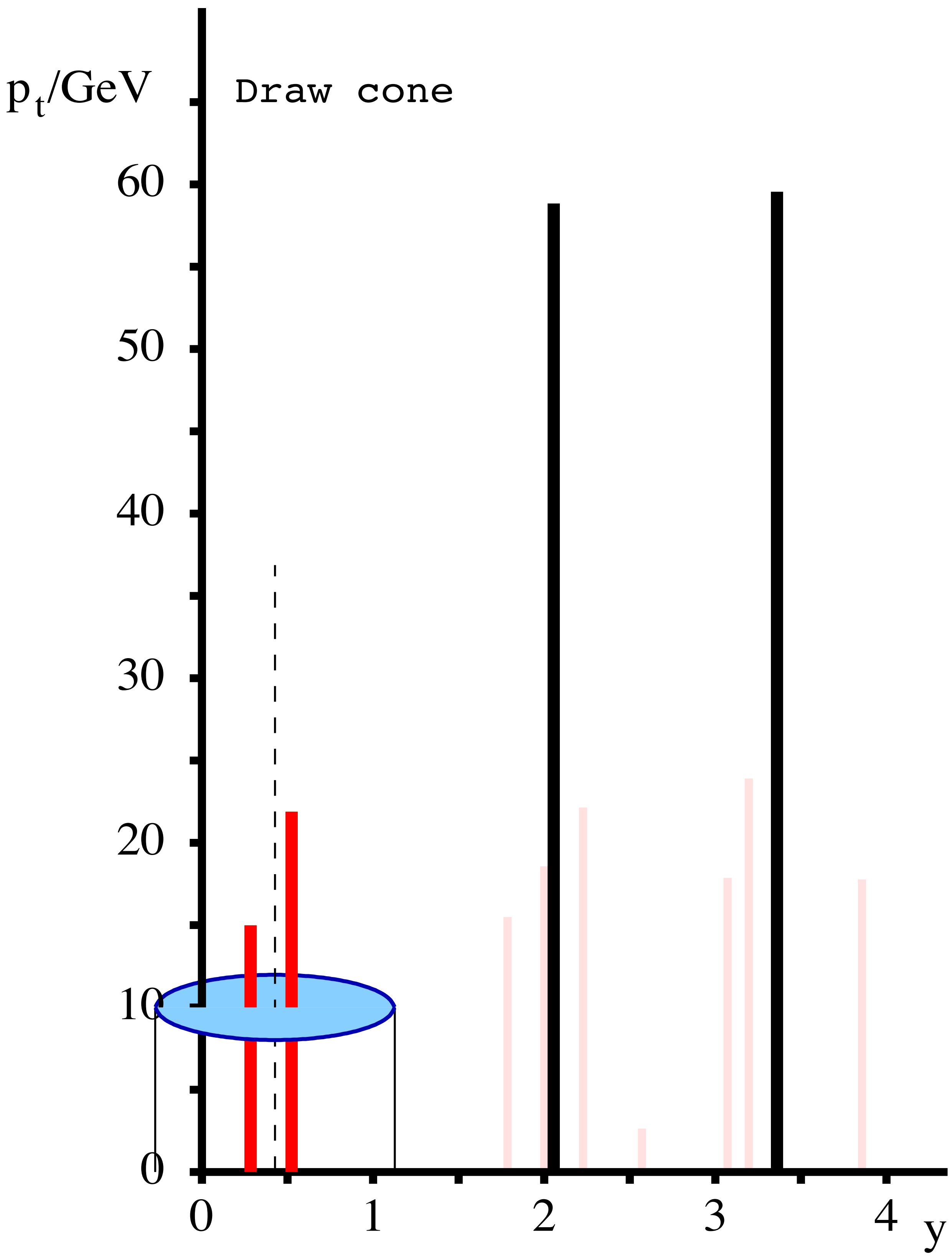
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

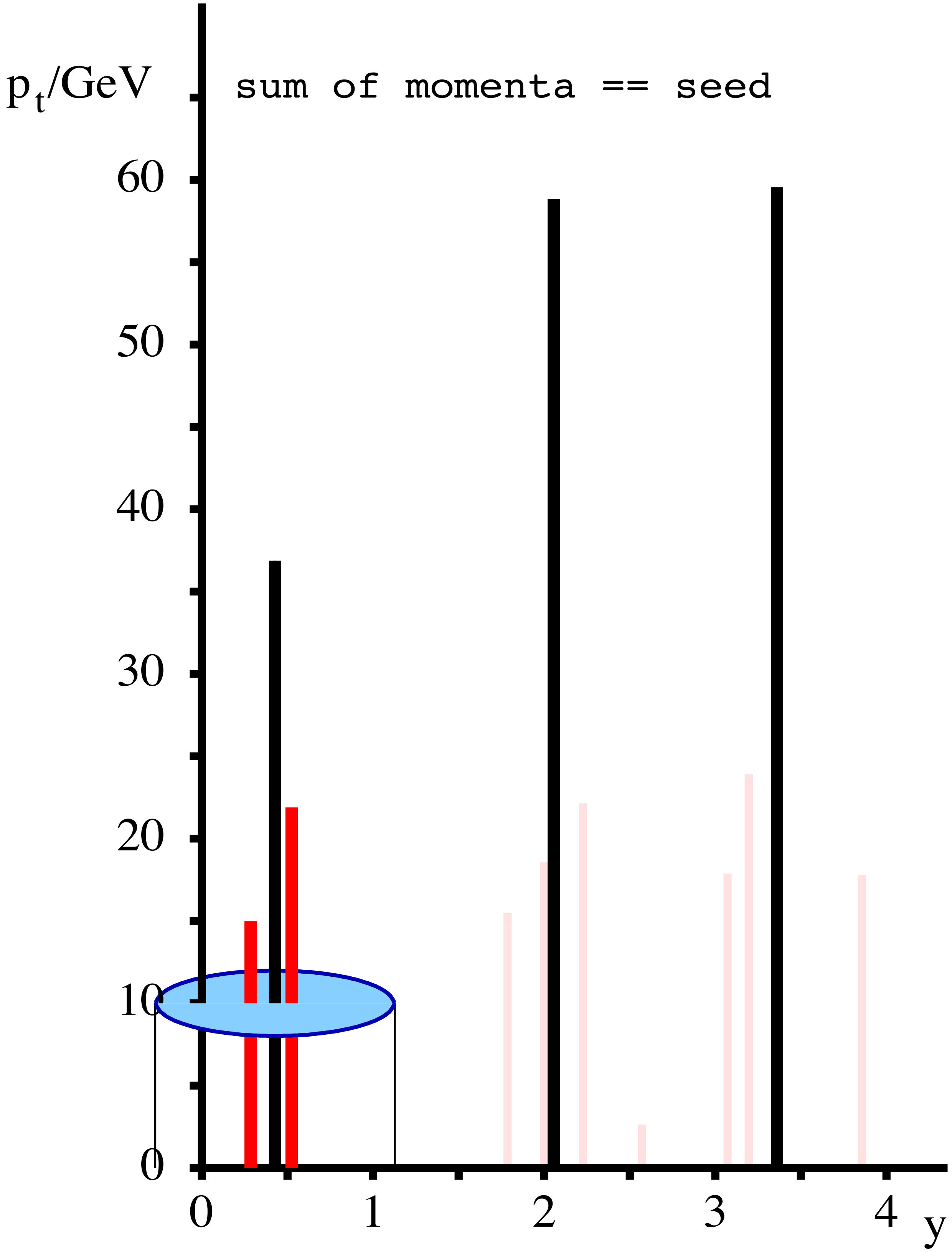
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

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



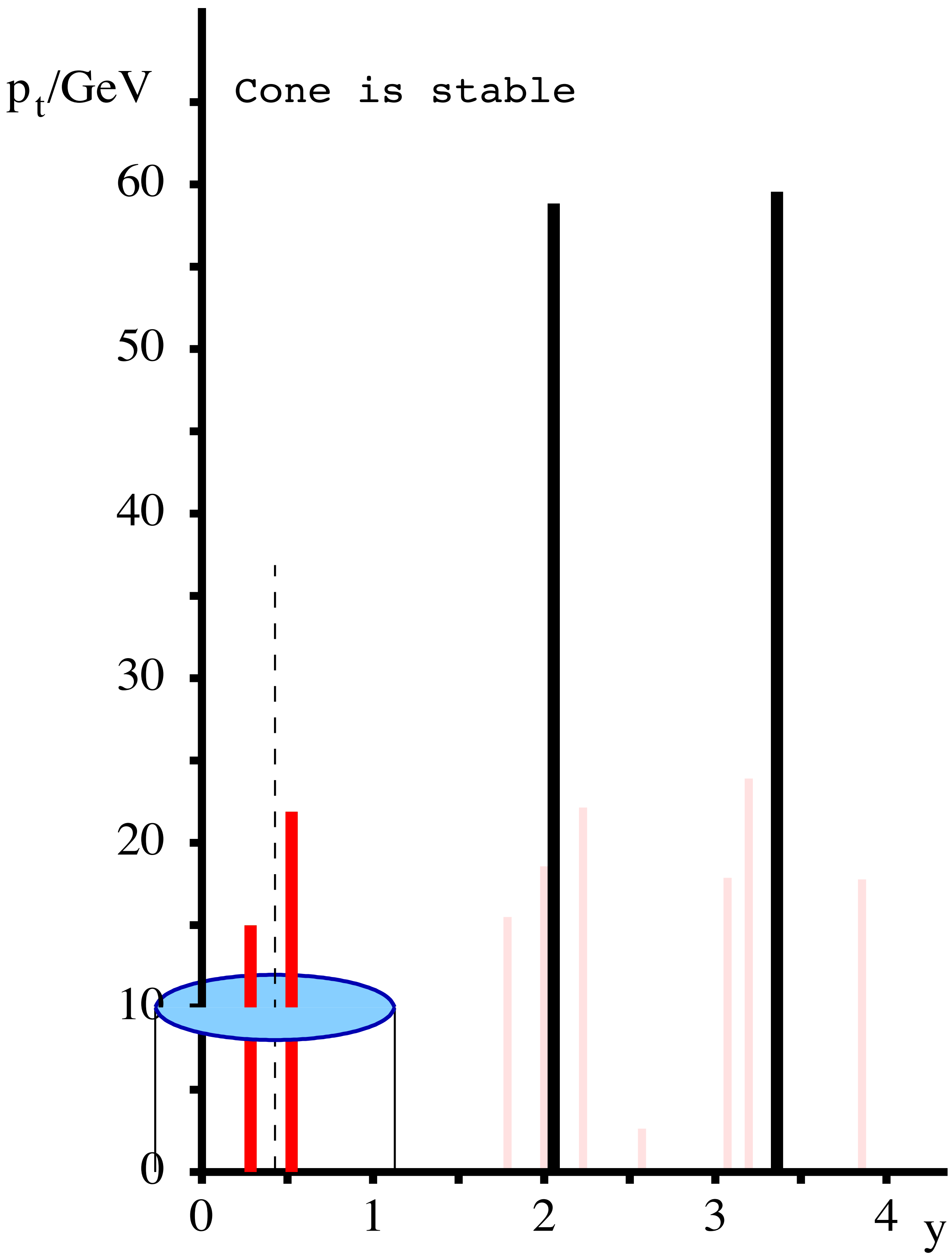
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

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



One of the simplest of the cone algs

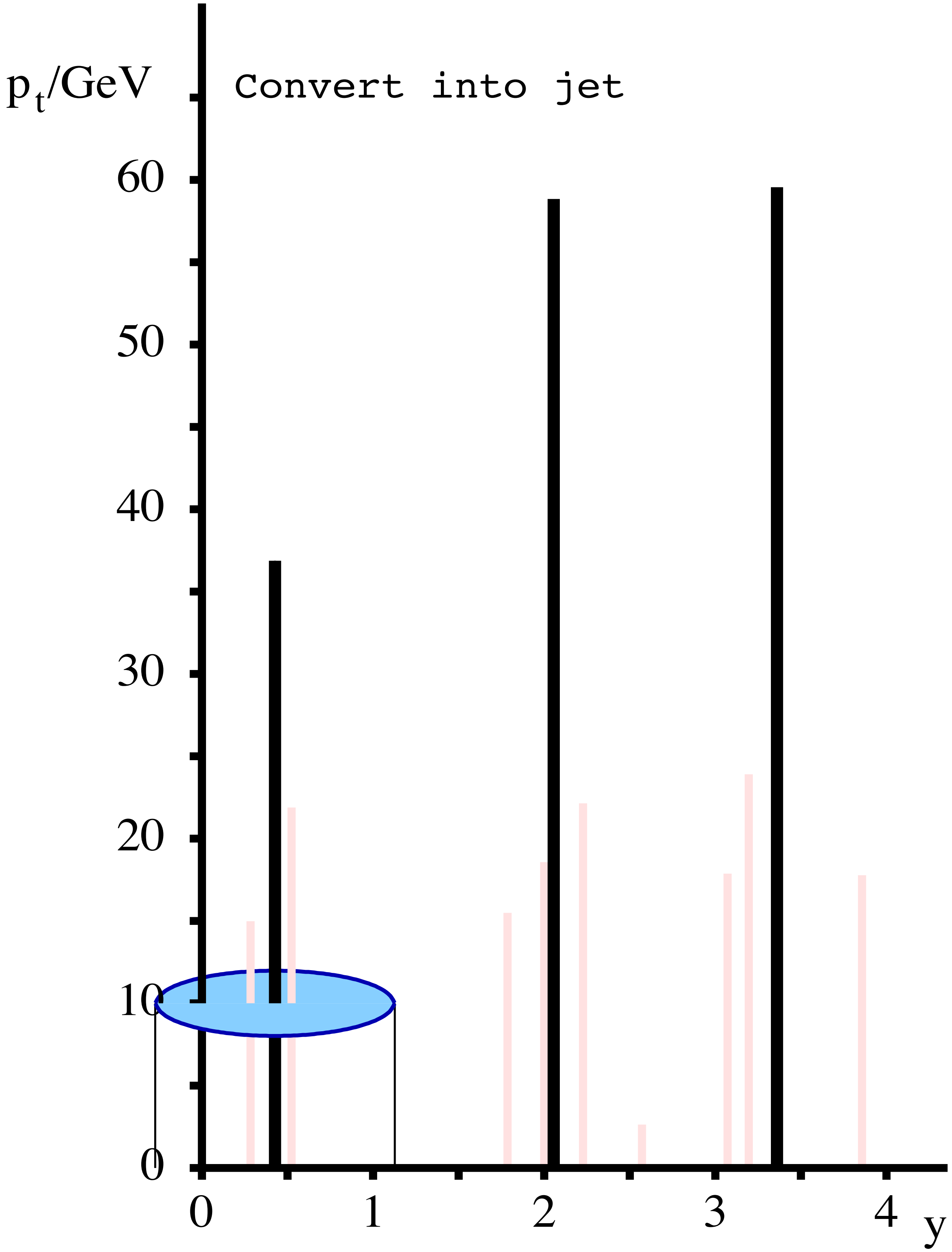
e.g. CMS iterative cone

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

Notes

- "Hardest particle" is collinear unsafe

more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

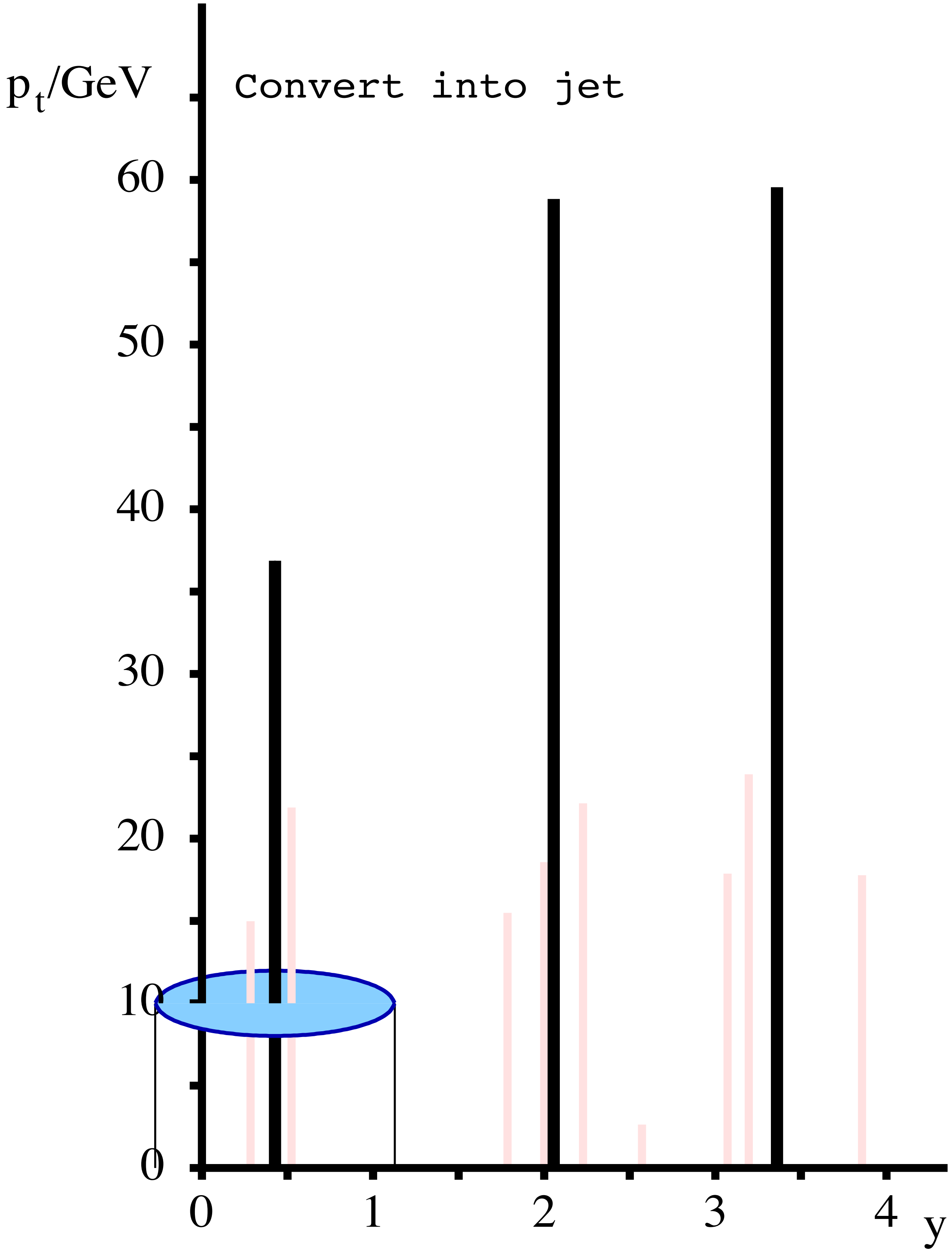
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe

more right away...

Iterative Cone, Prog Removal (IC-PR)



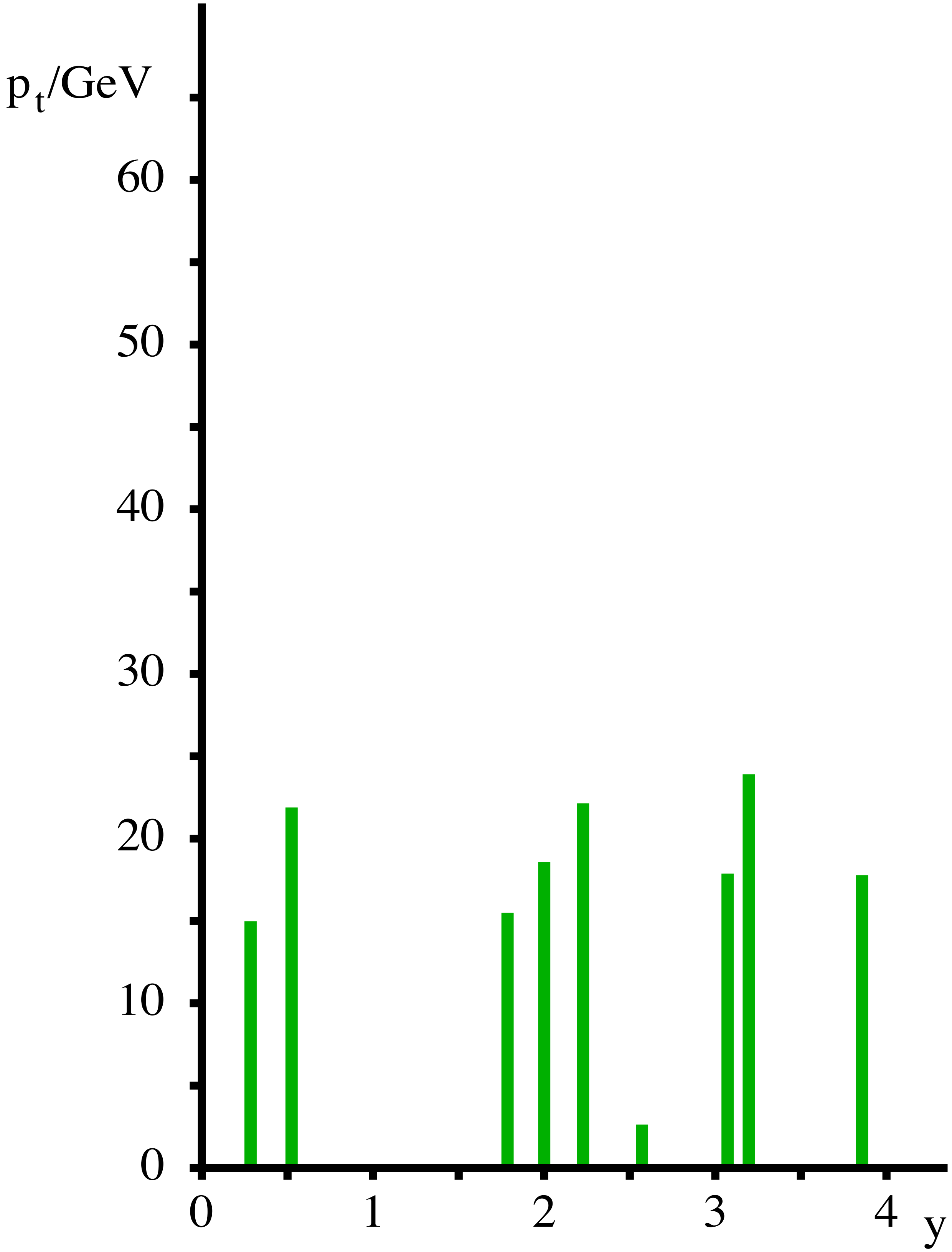
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

- "Hardest particle" is collinear unsafe more right away...



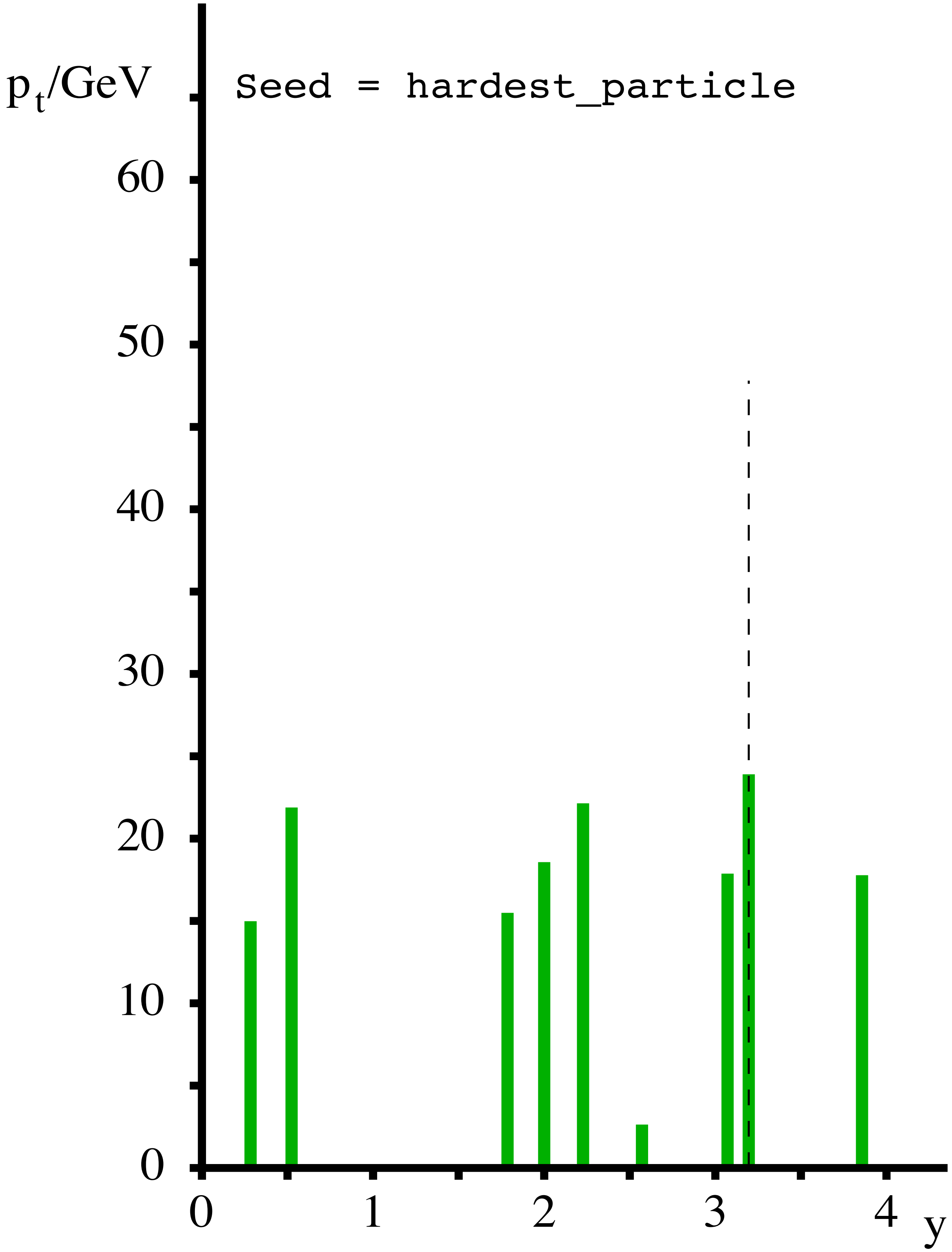
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

- "Hardest particle" is collinear unsafe more right away...



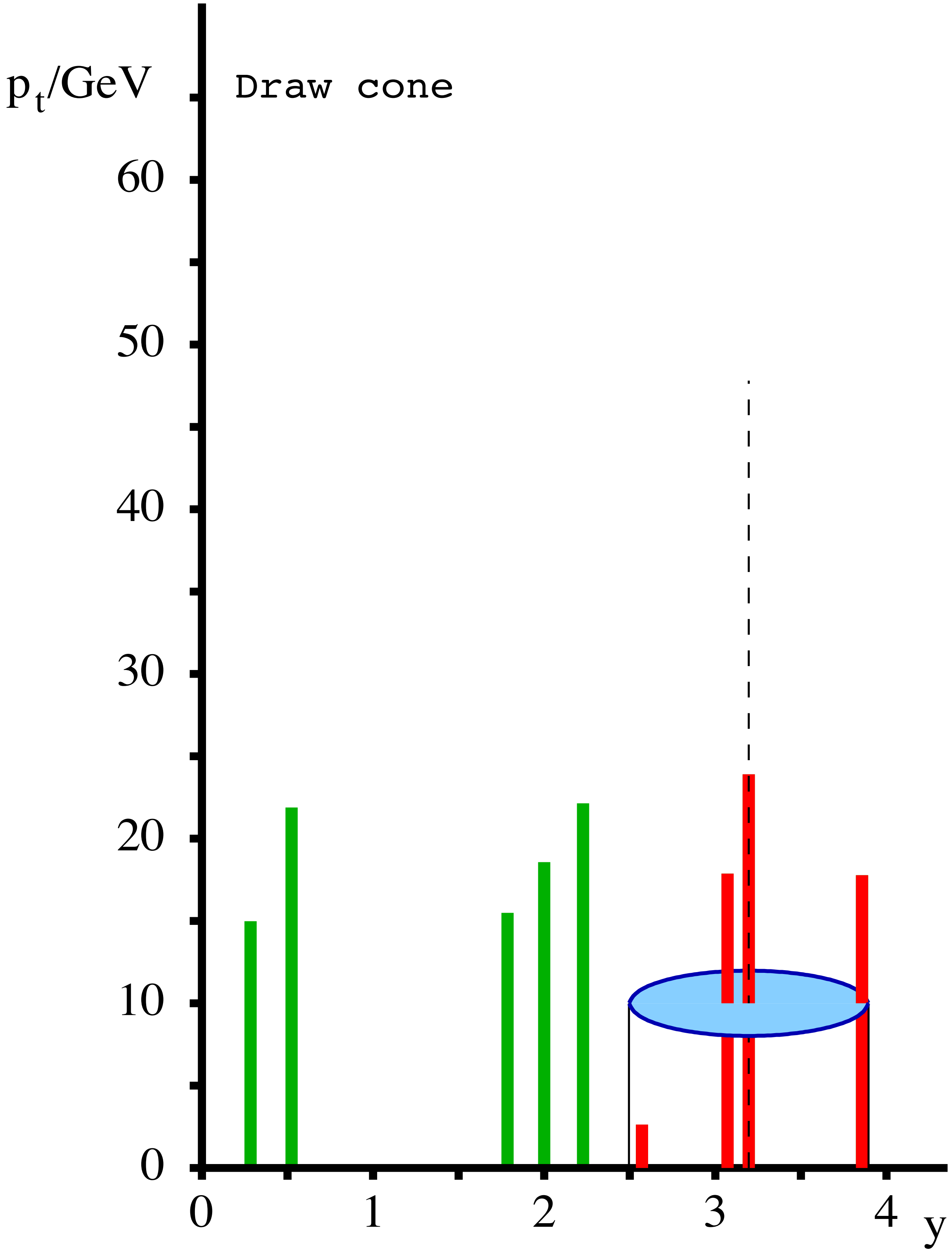
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

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



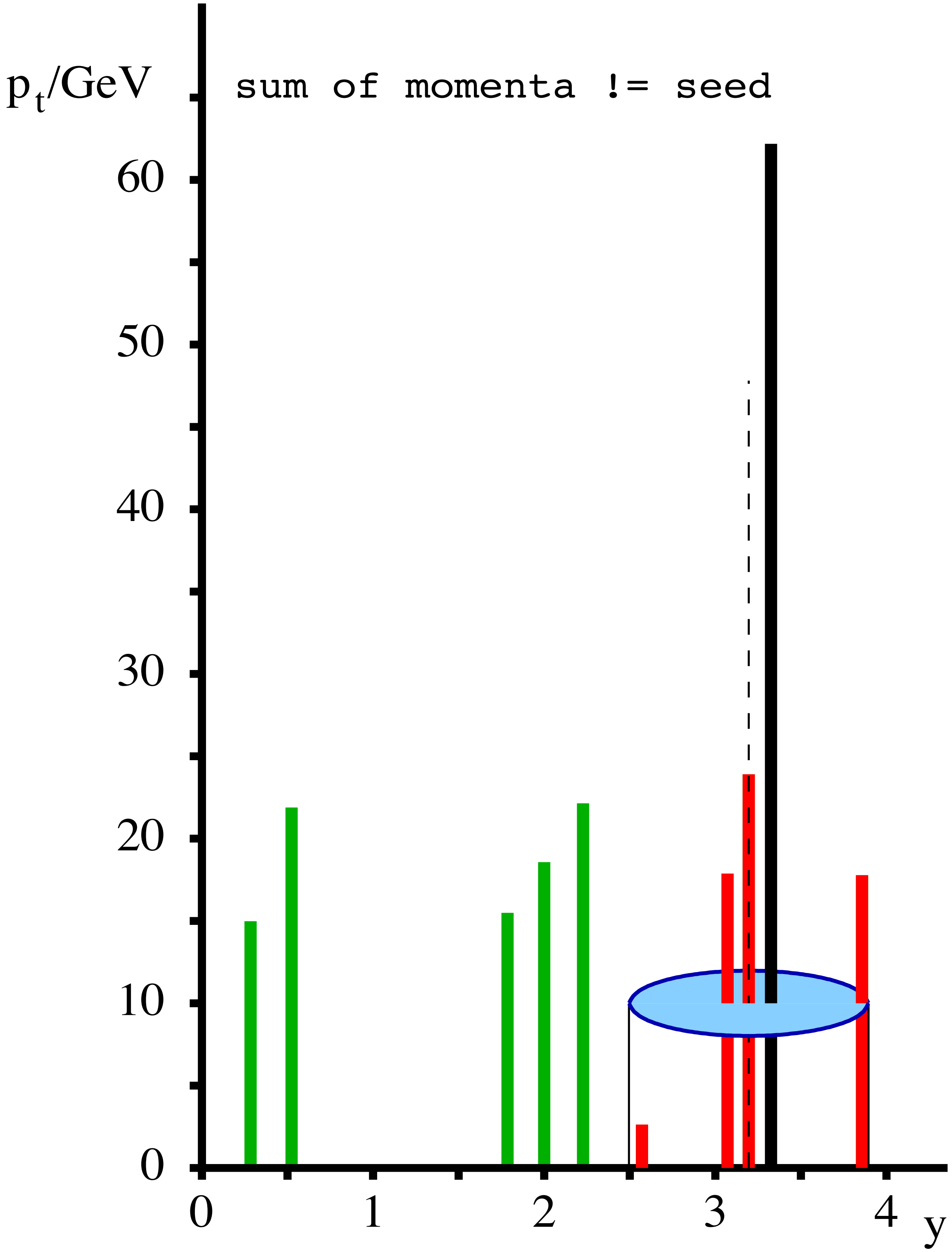
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

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



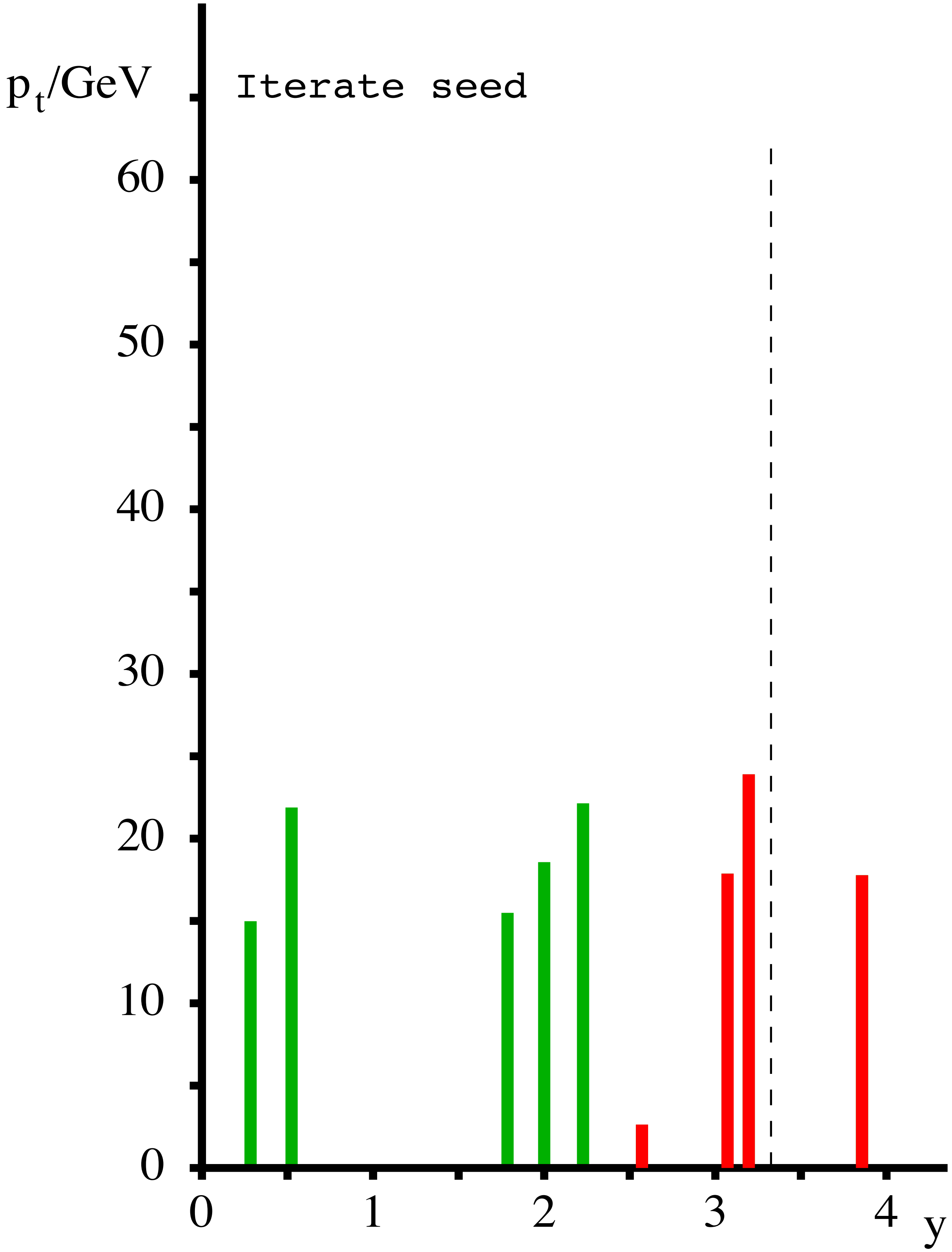
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

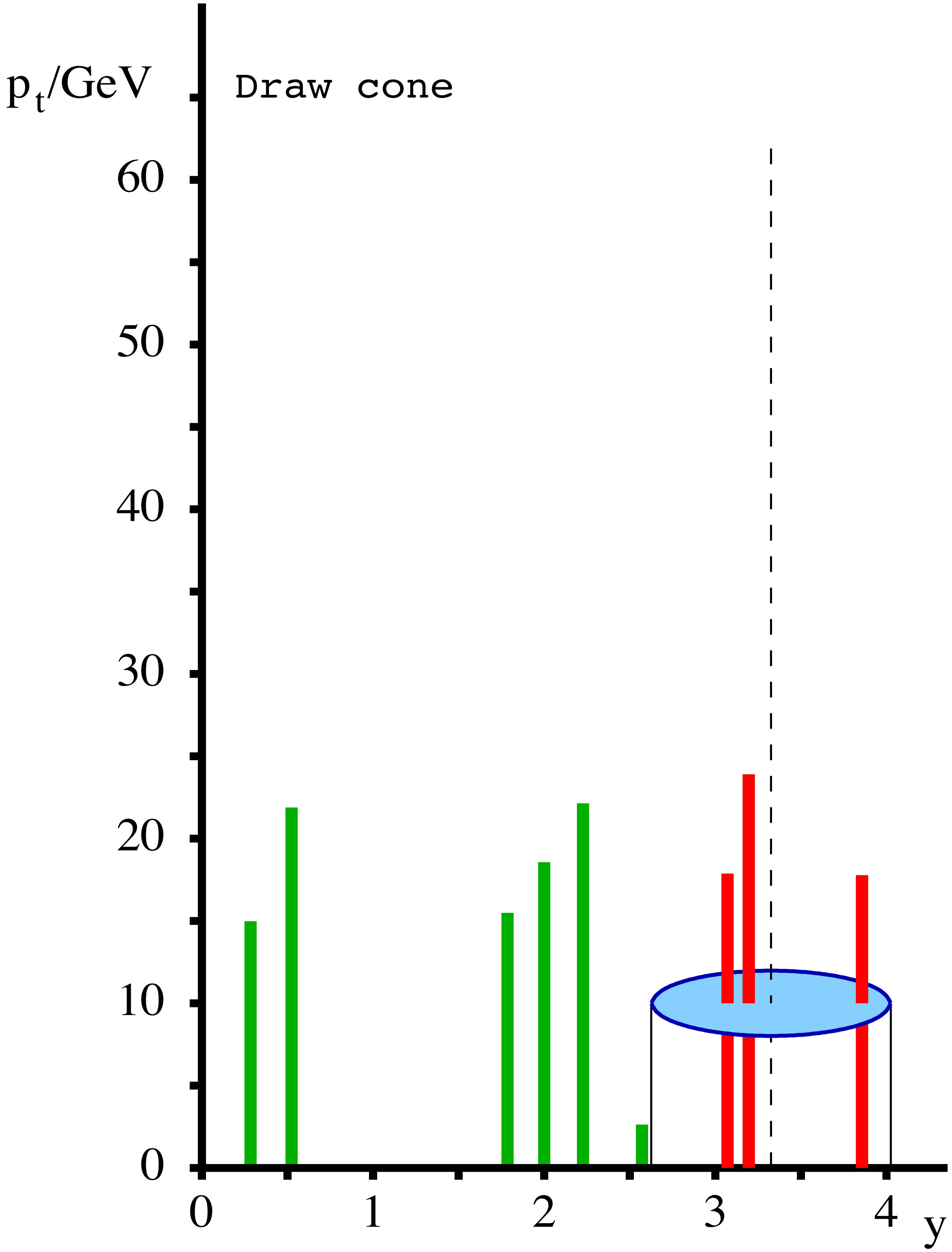
- "Hardest particle" is collinear unsafe more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

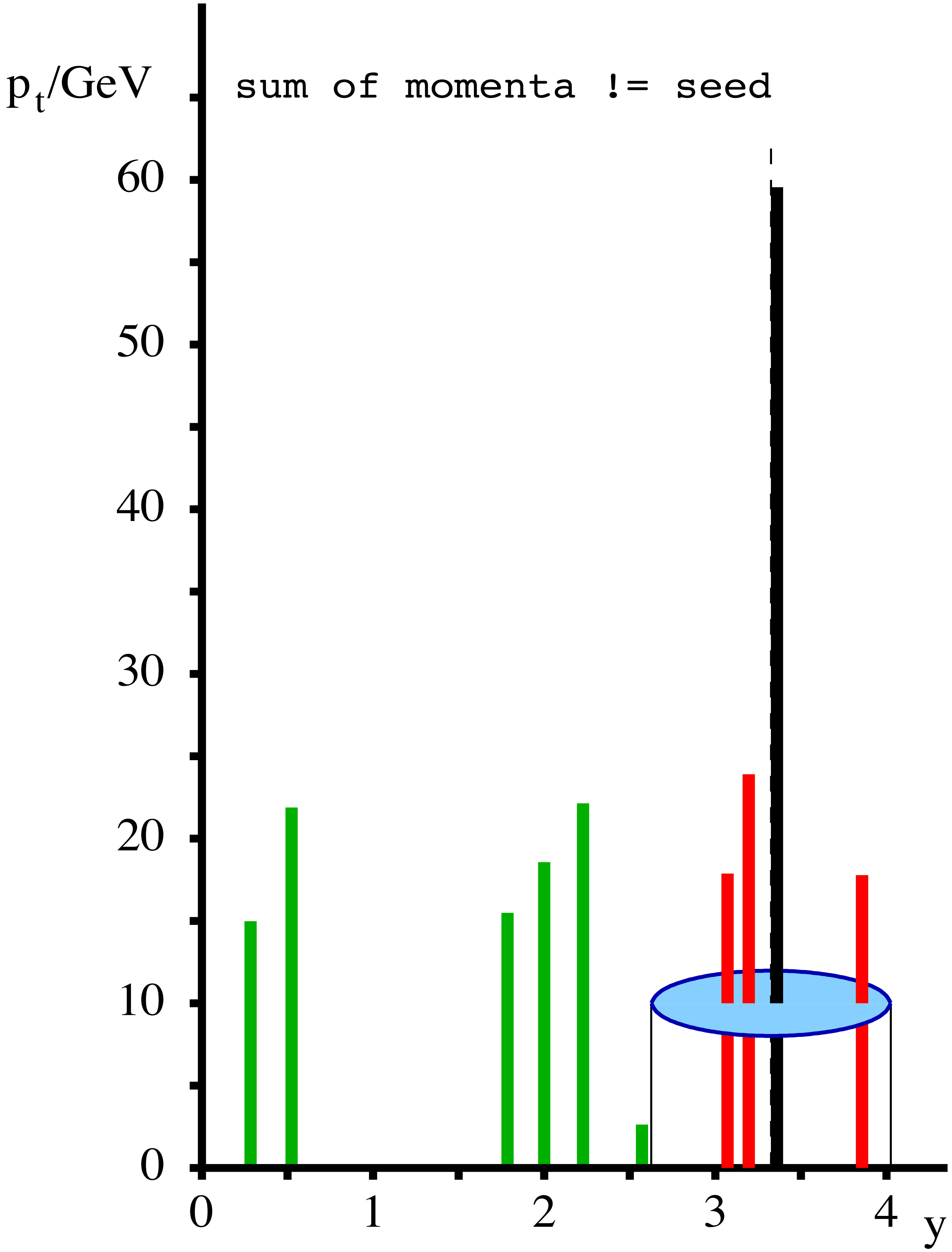
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable



One of the simplest of the cone algs

e.g. CMS iterative cone

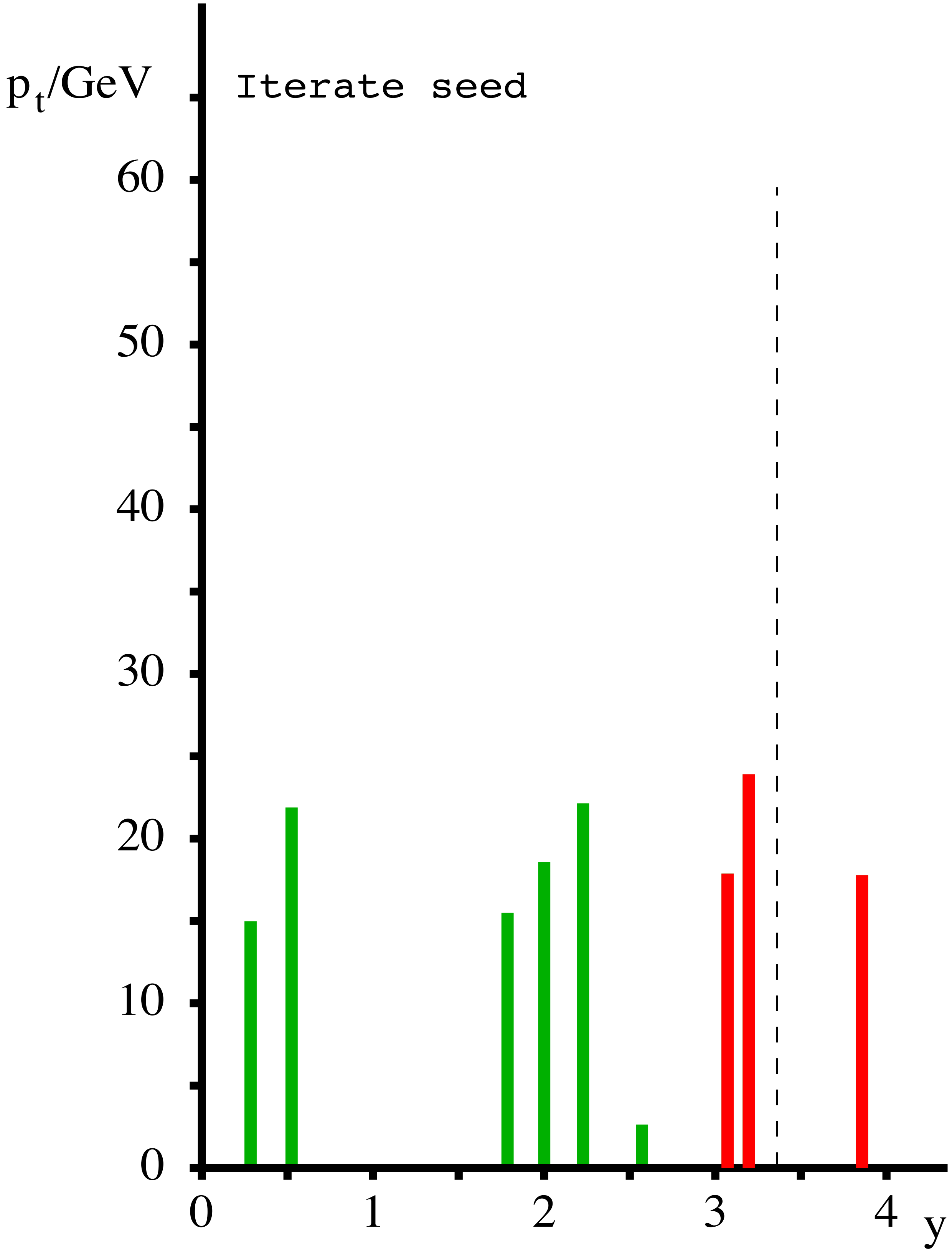
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable



One of the simplest of the cone algs

e.g. CMS iterative cone

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



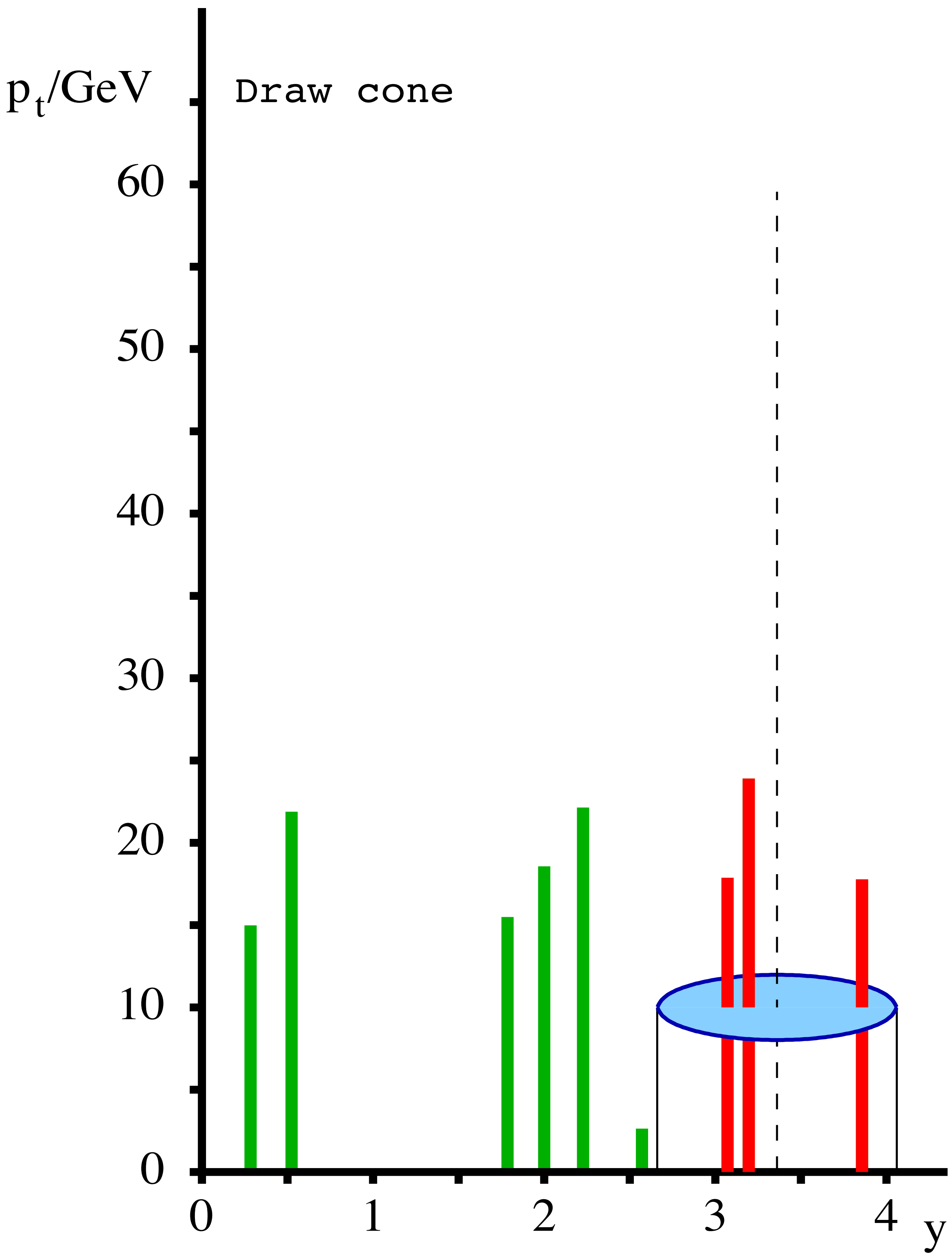
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

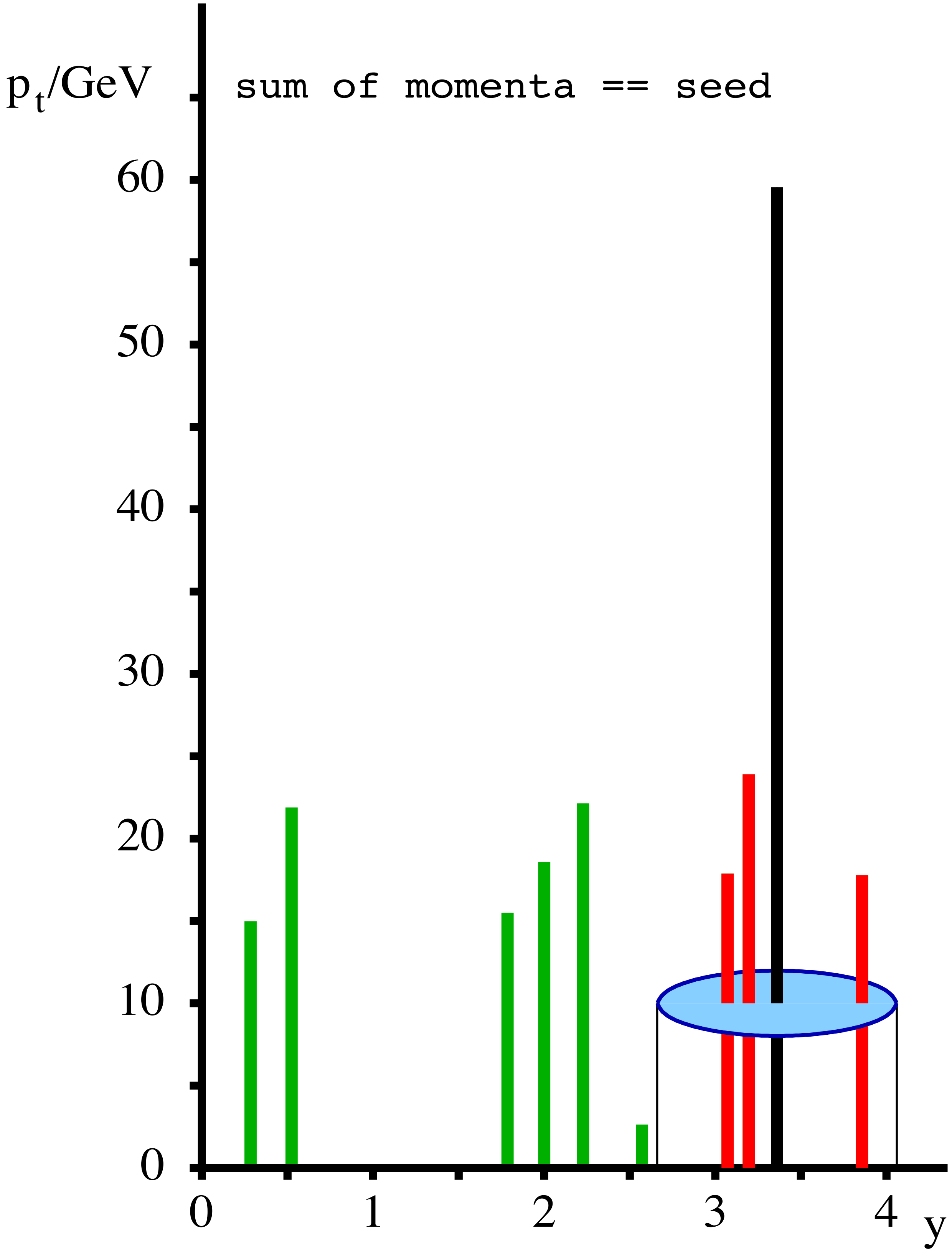
- "Hardest particle" is collinear unsafe more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

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



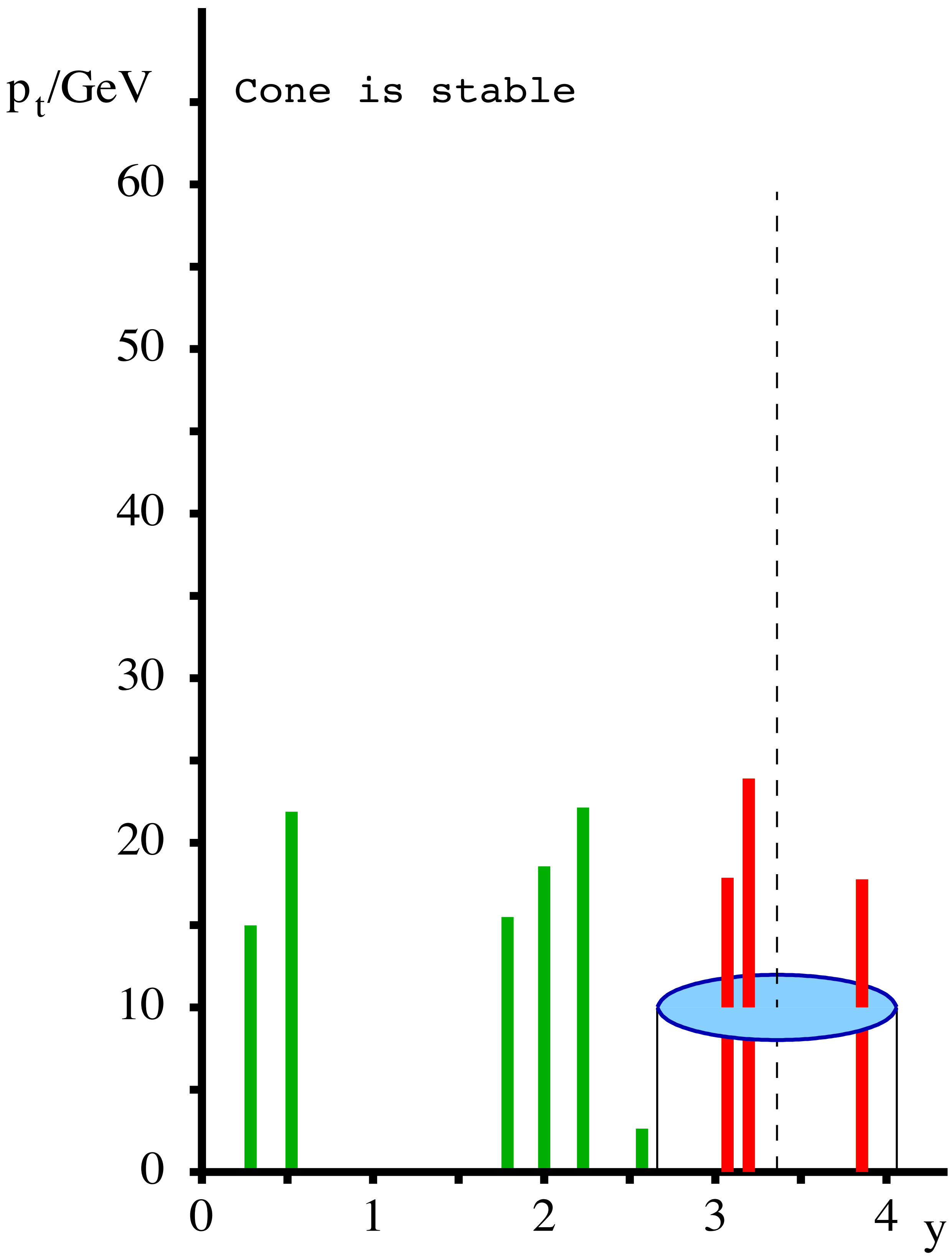
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

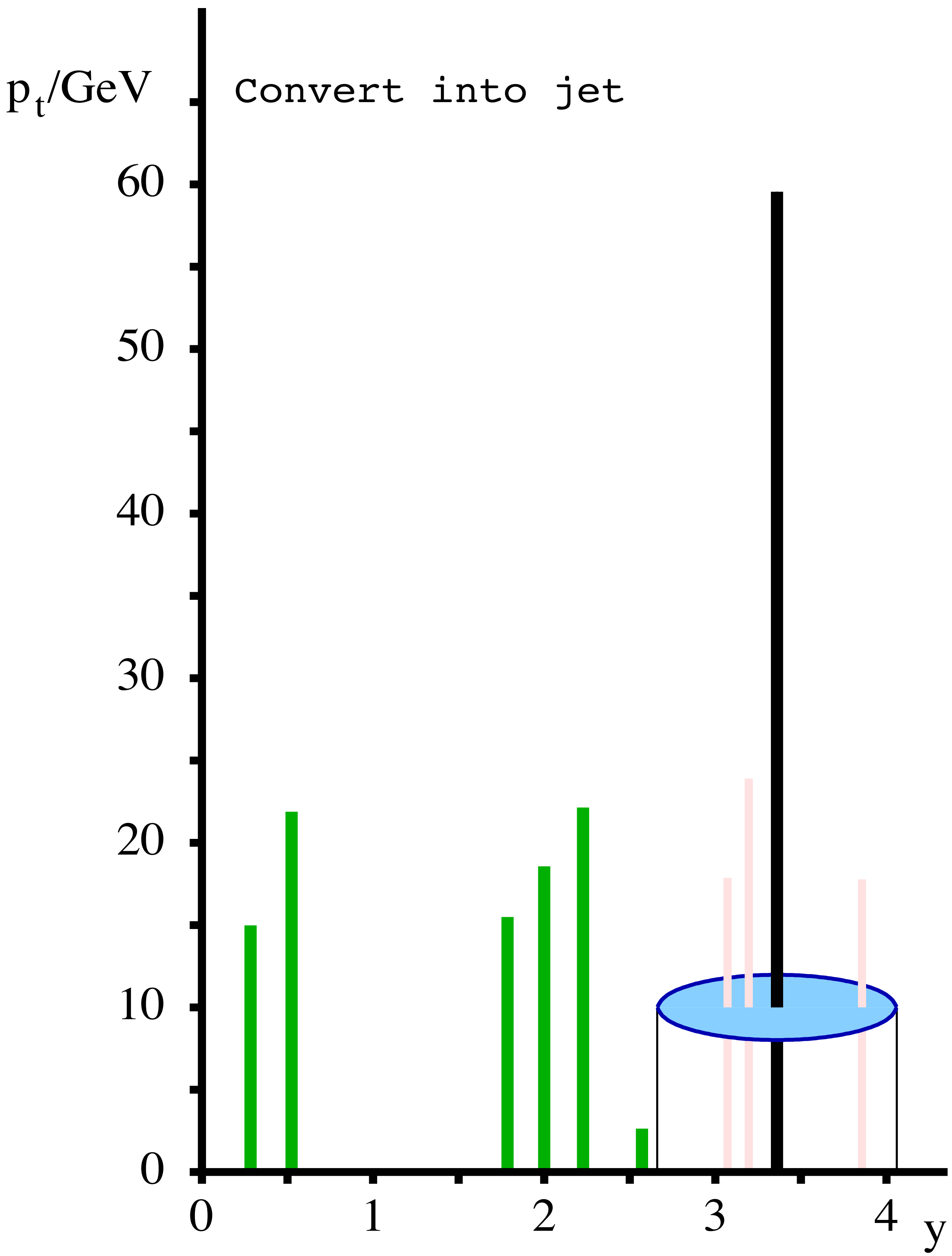
- “Hardest particle” is collinear unsafe more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

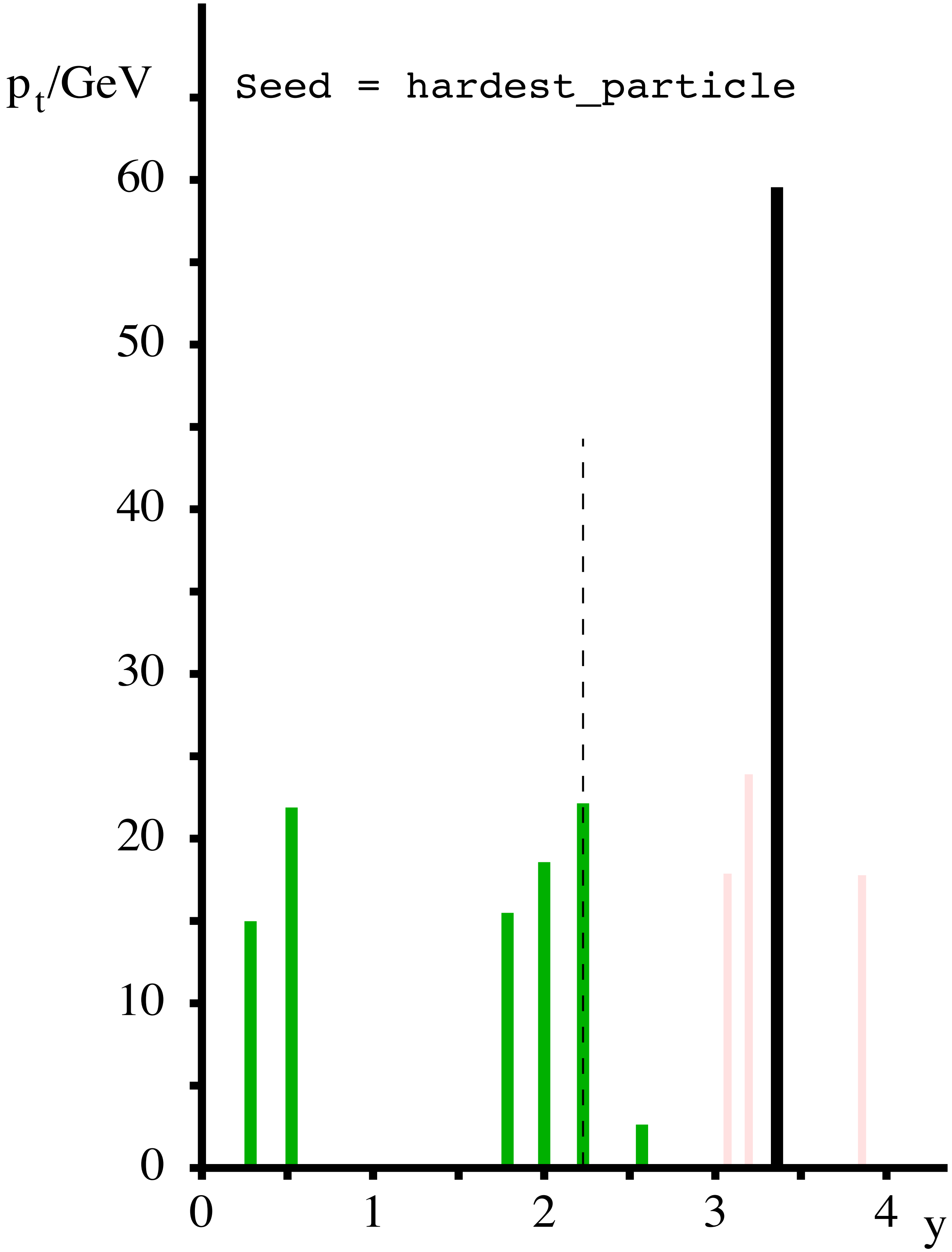
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

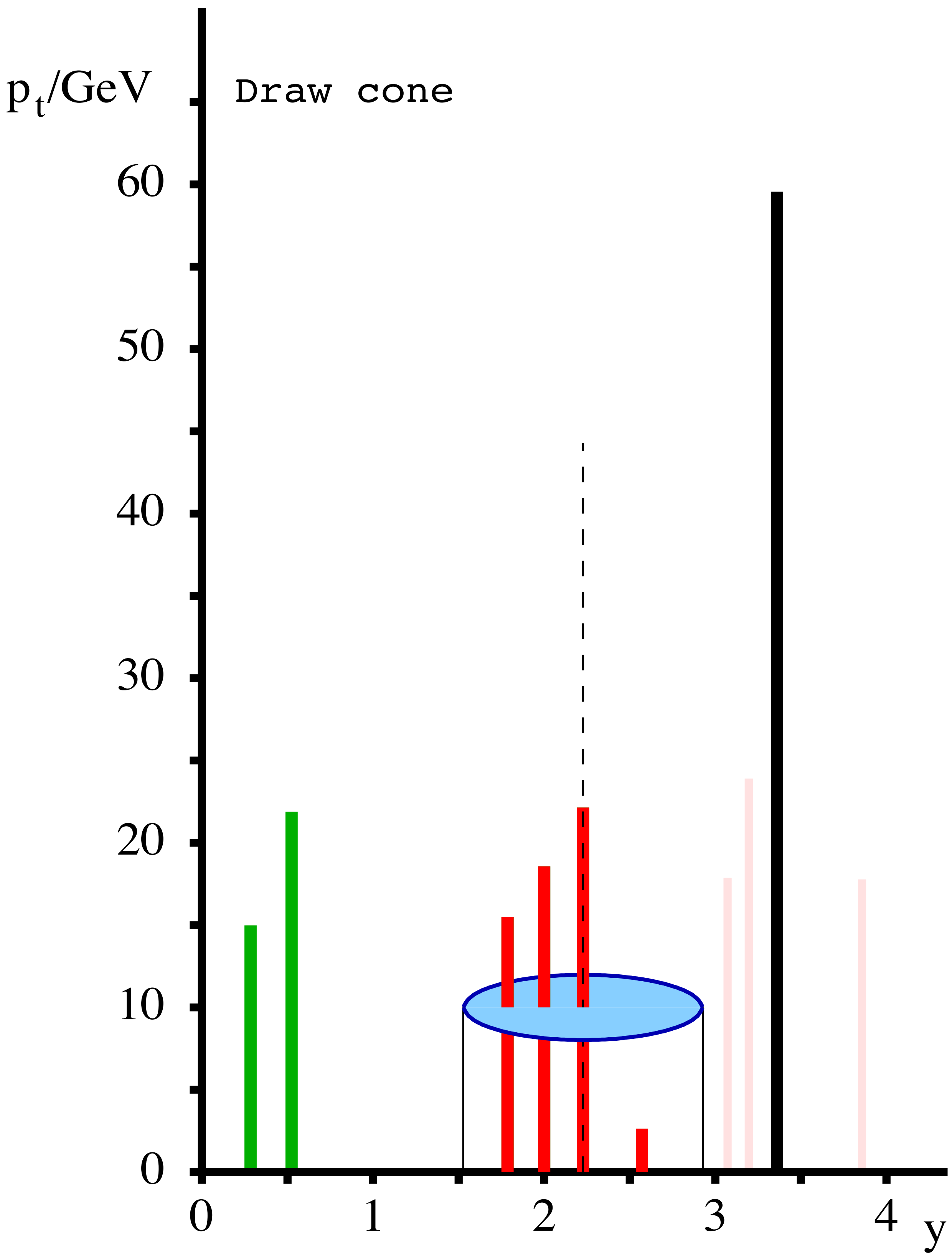
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

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



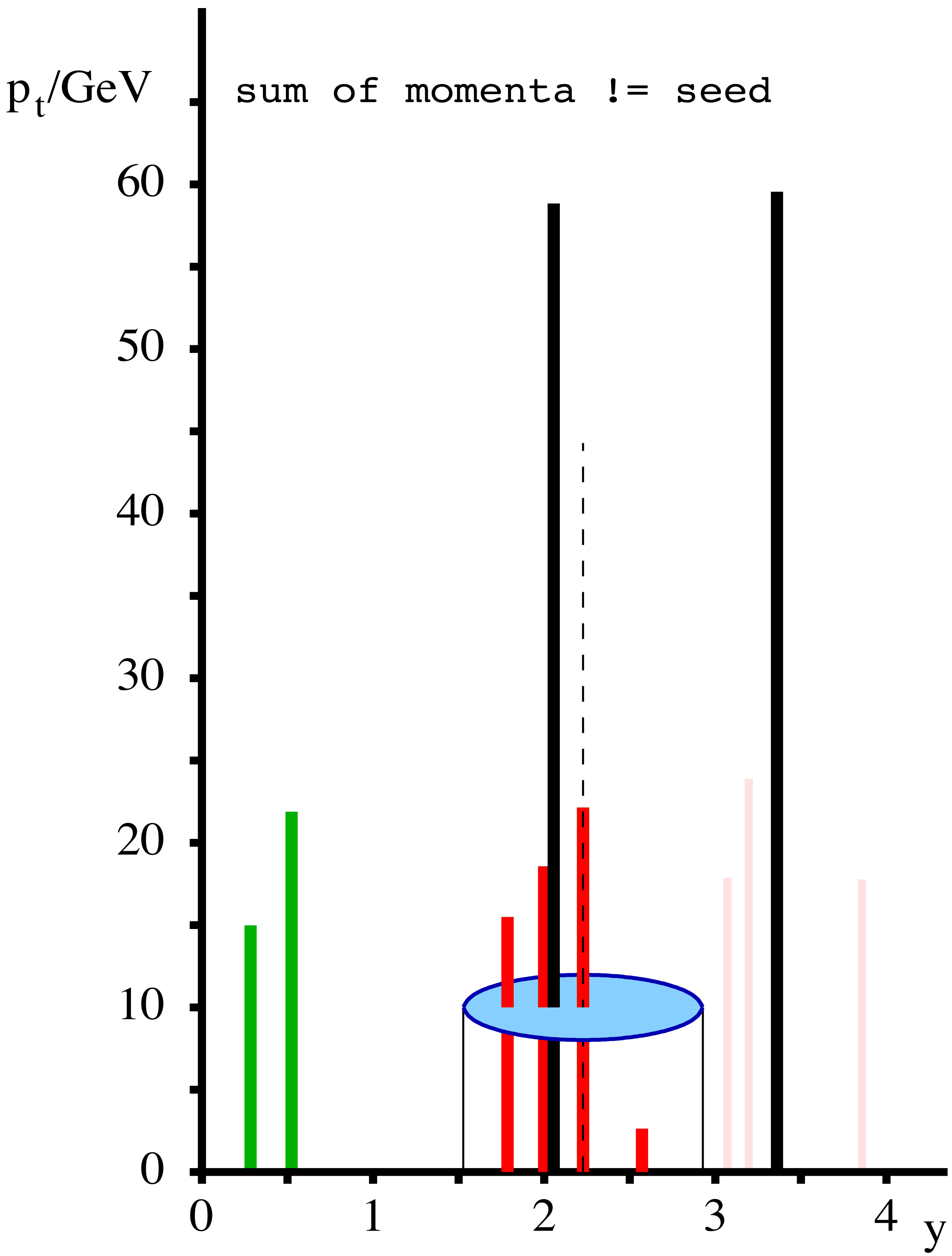
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

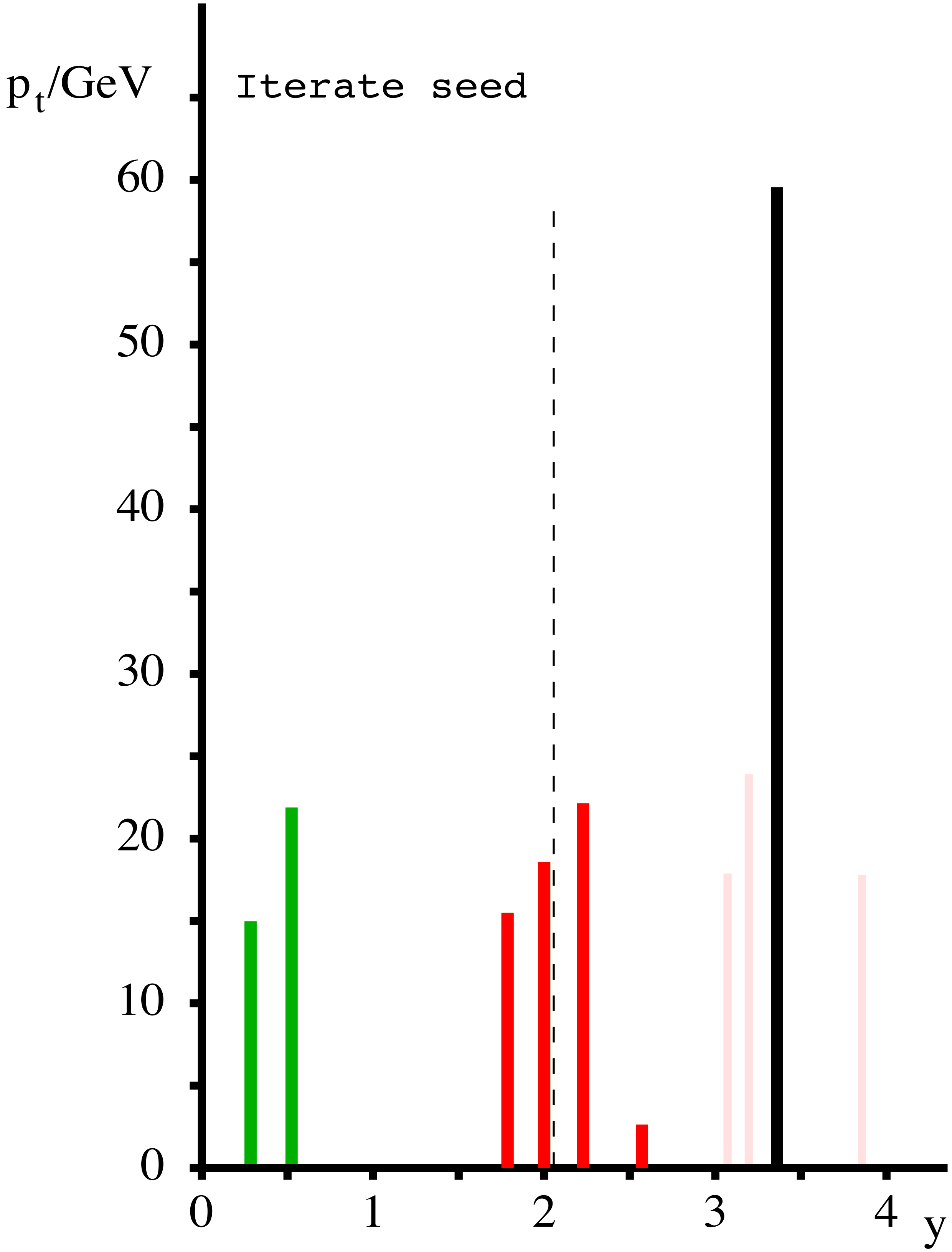
- "Hardest particle" is collinear unsafe more right away...



One of the simplest of the cone algs

e.g. CMS iterative cone

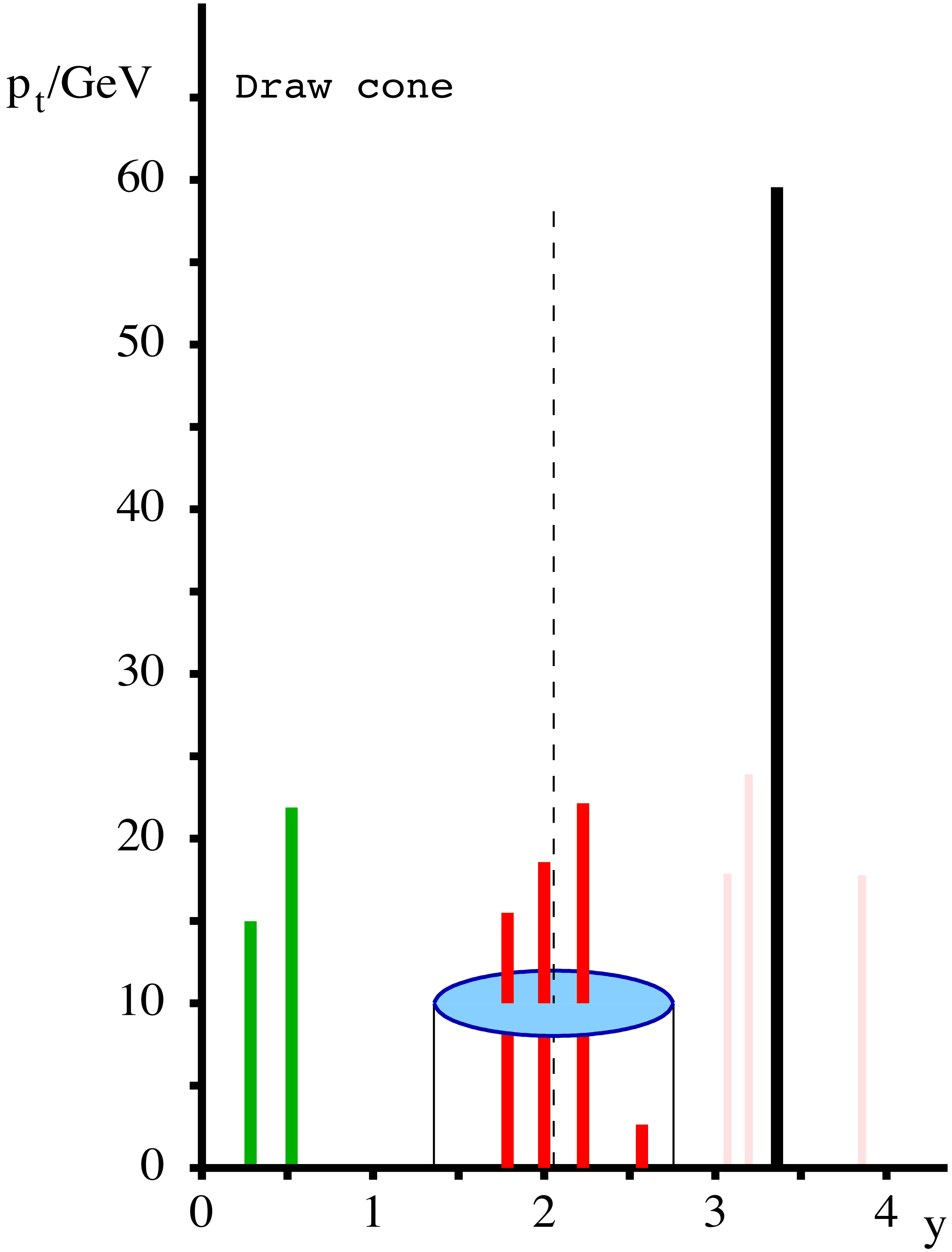
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

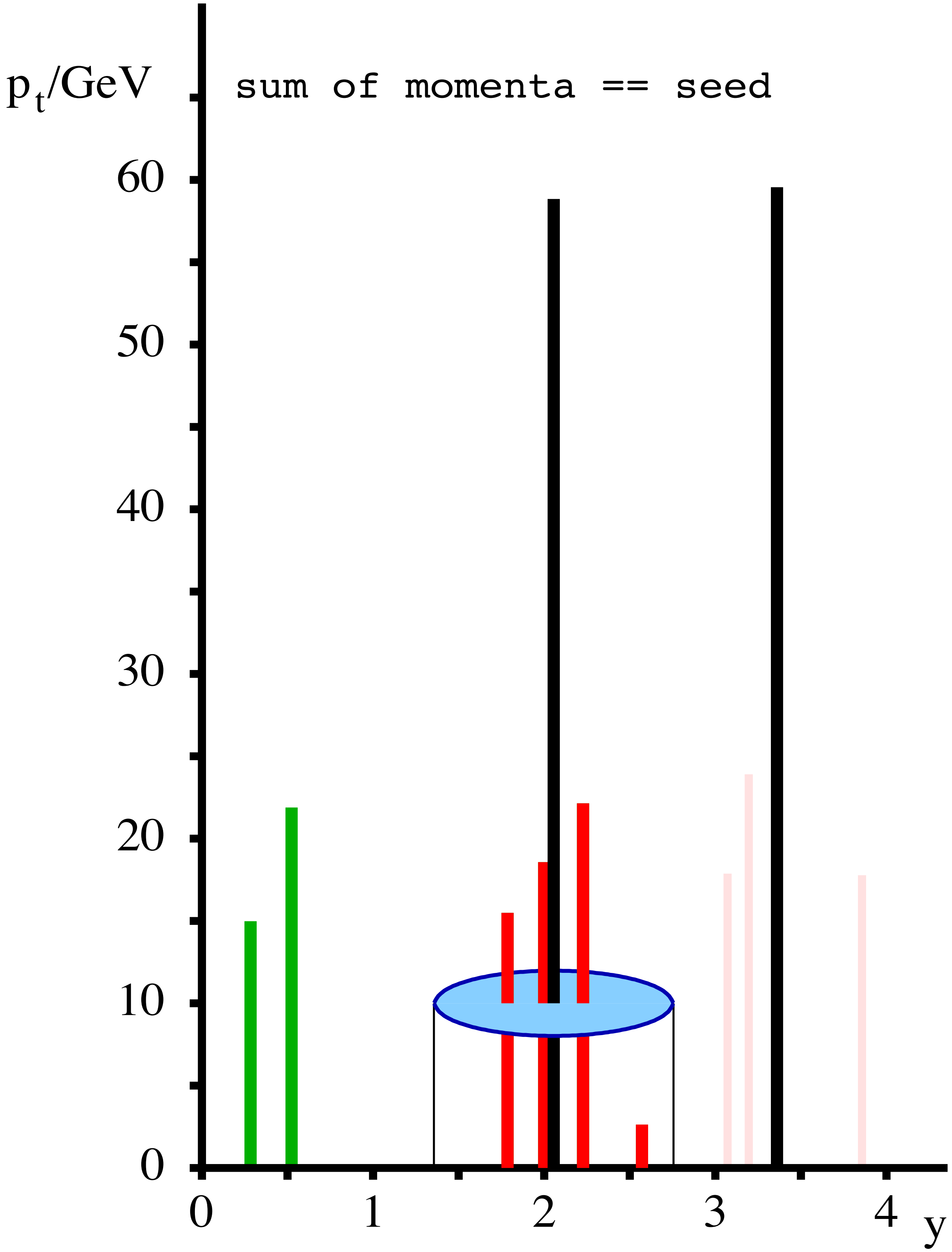
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

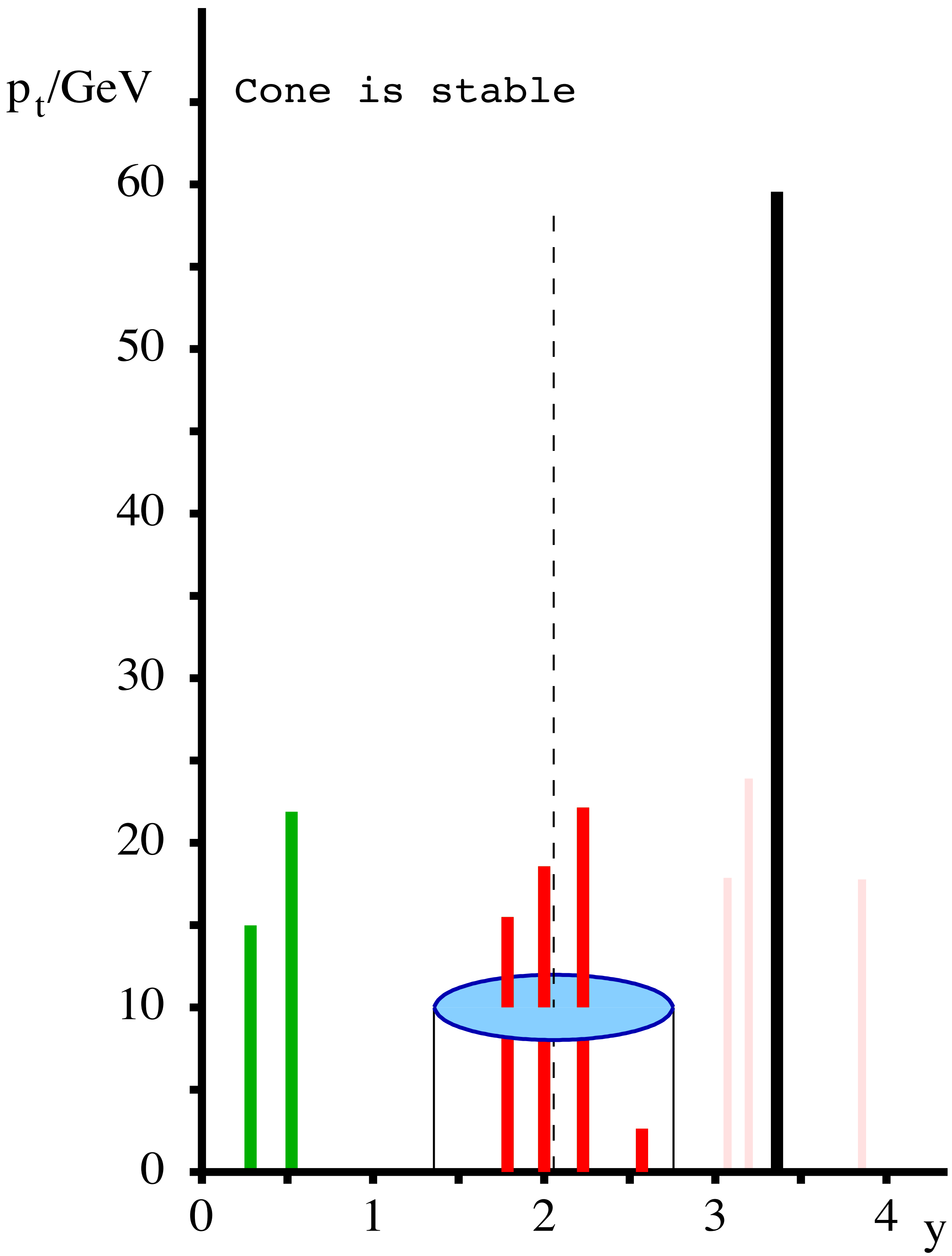
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

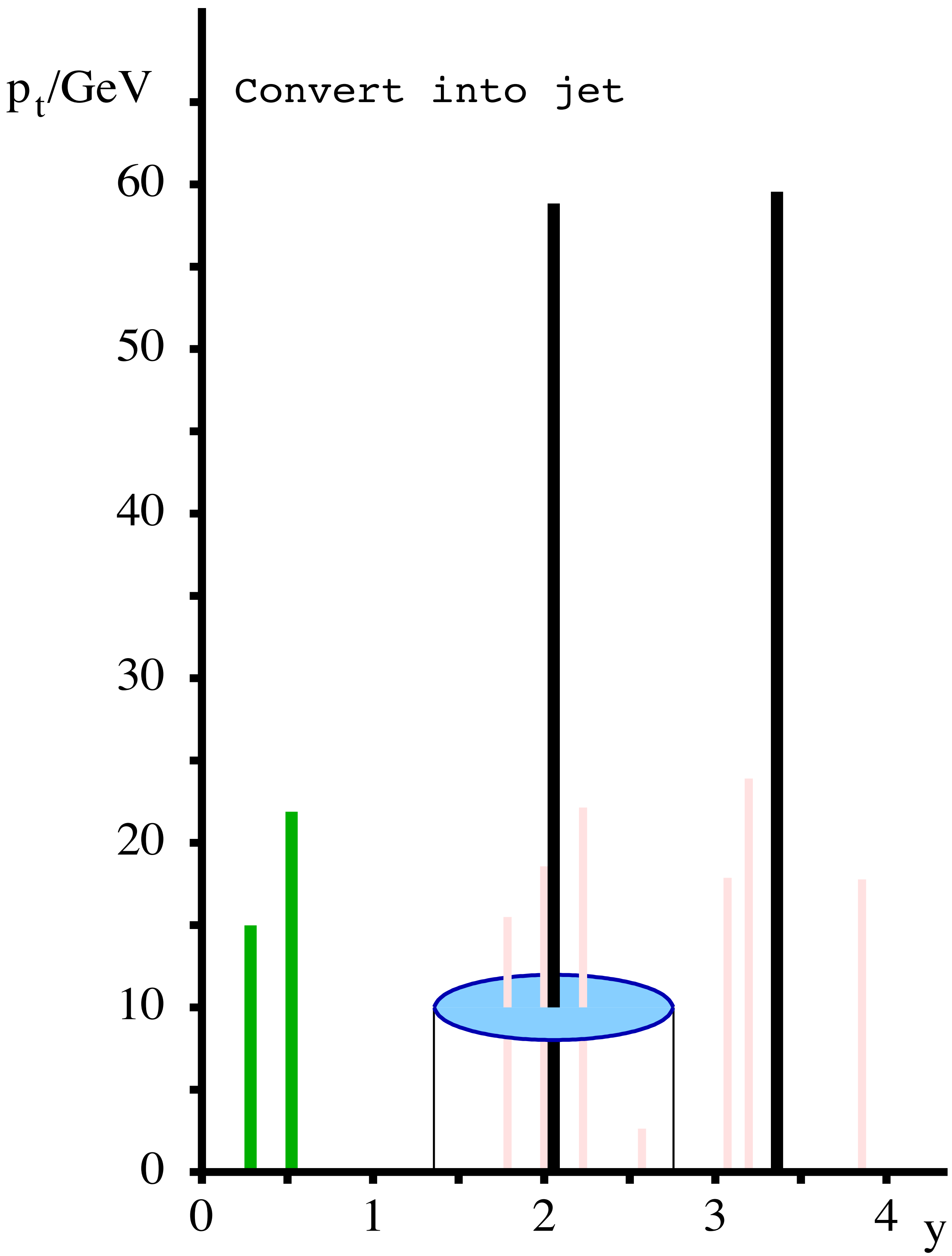
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

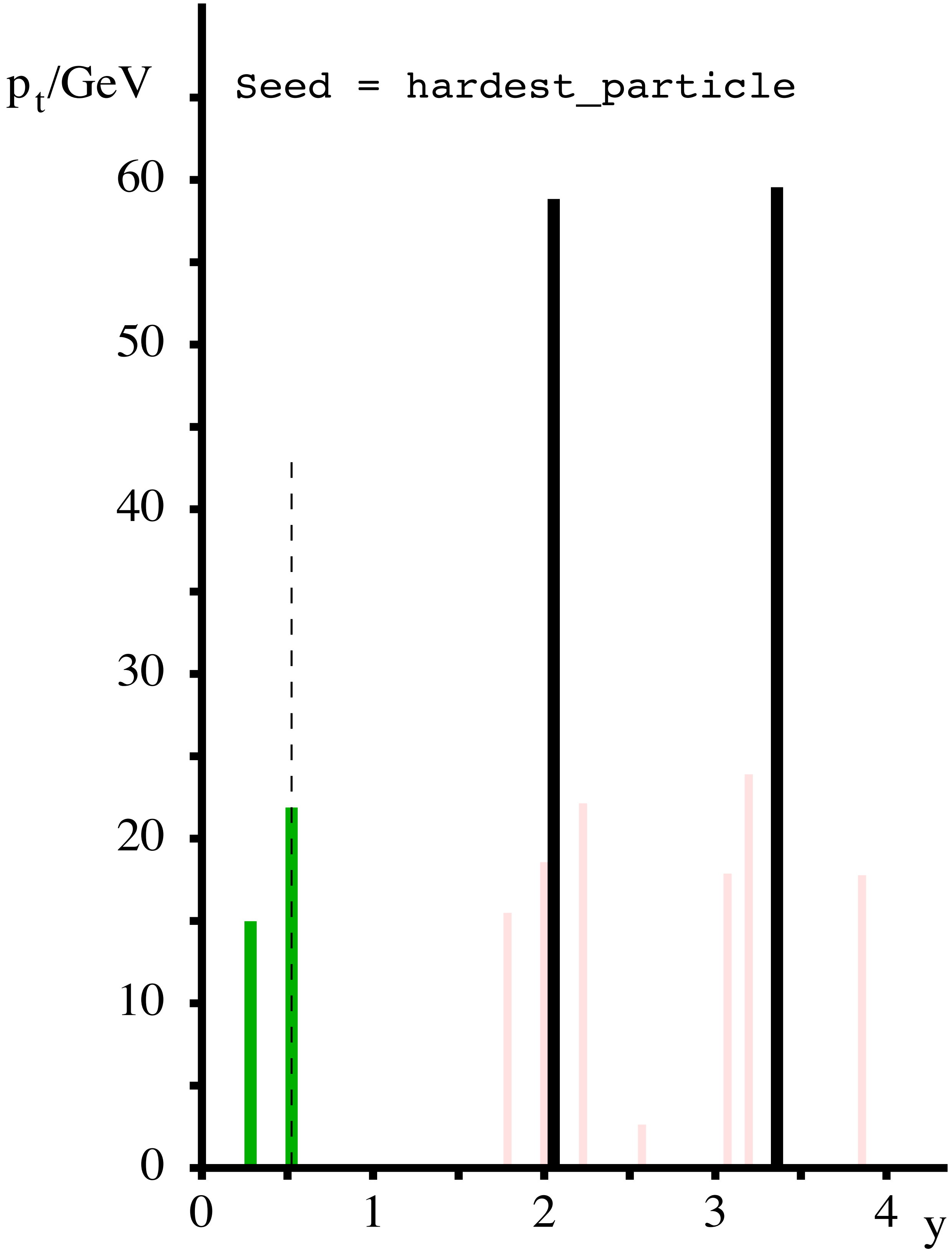
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

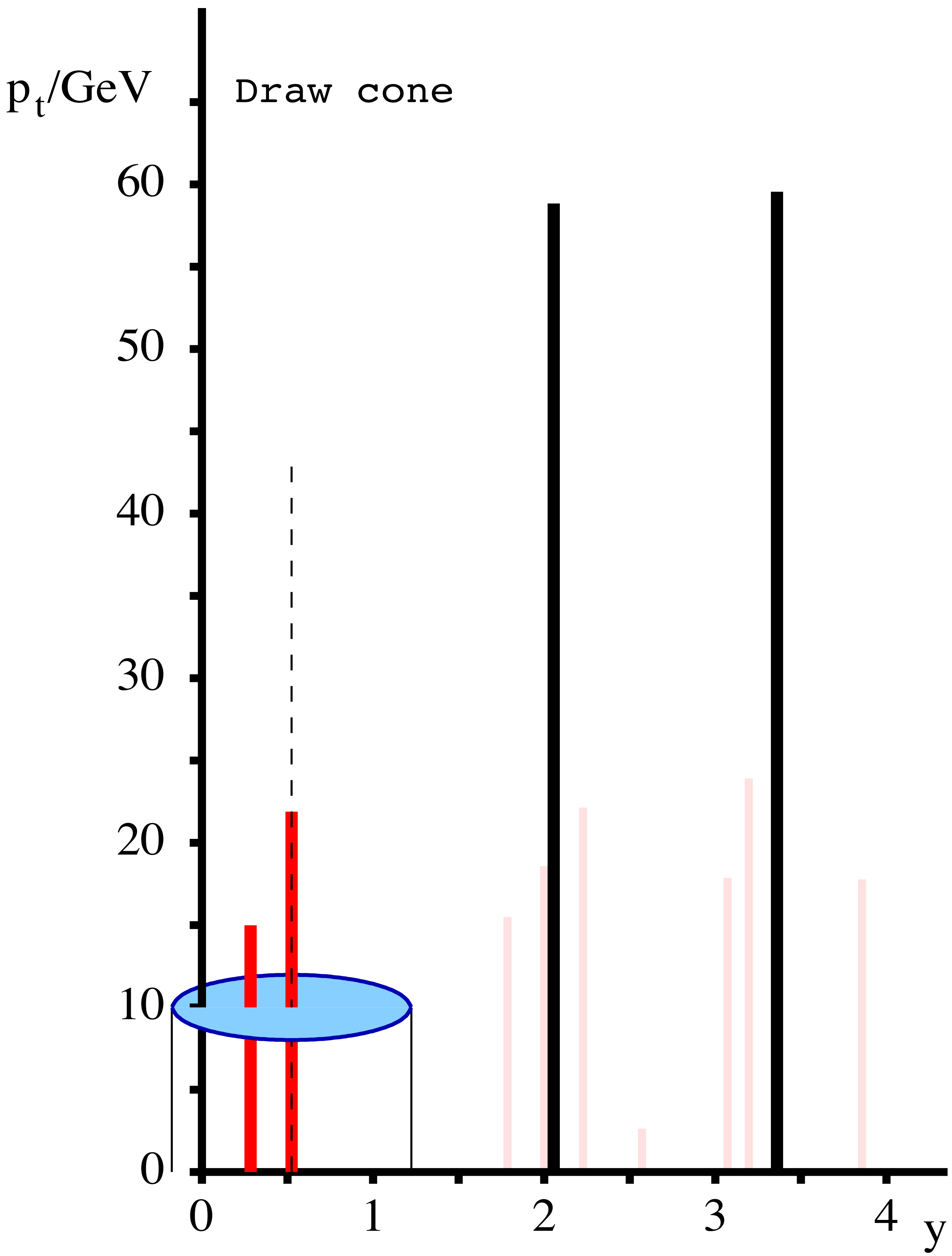
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

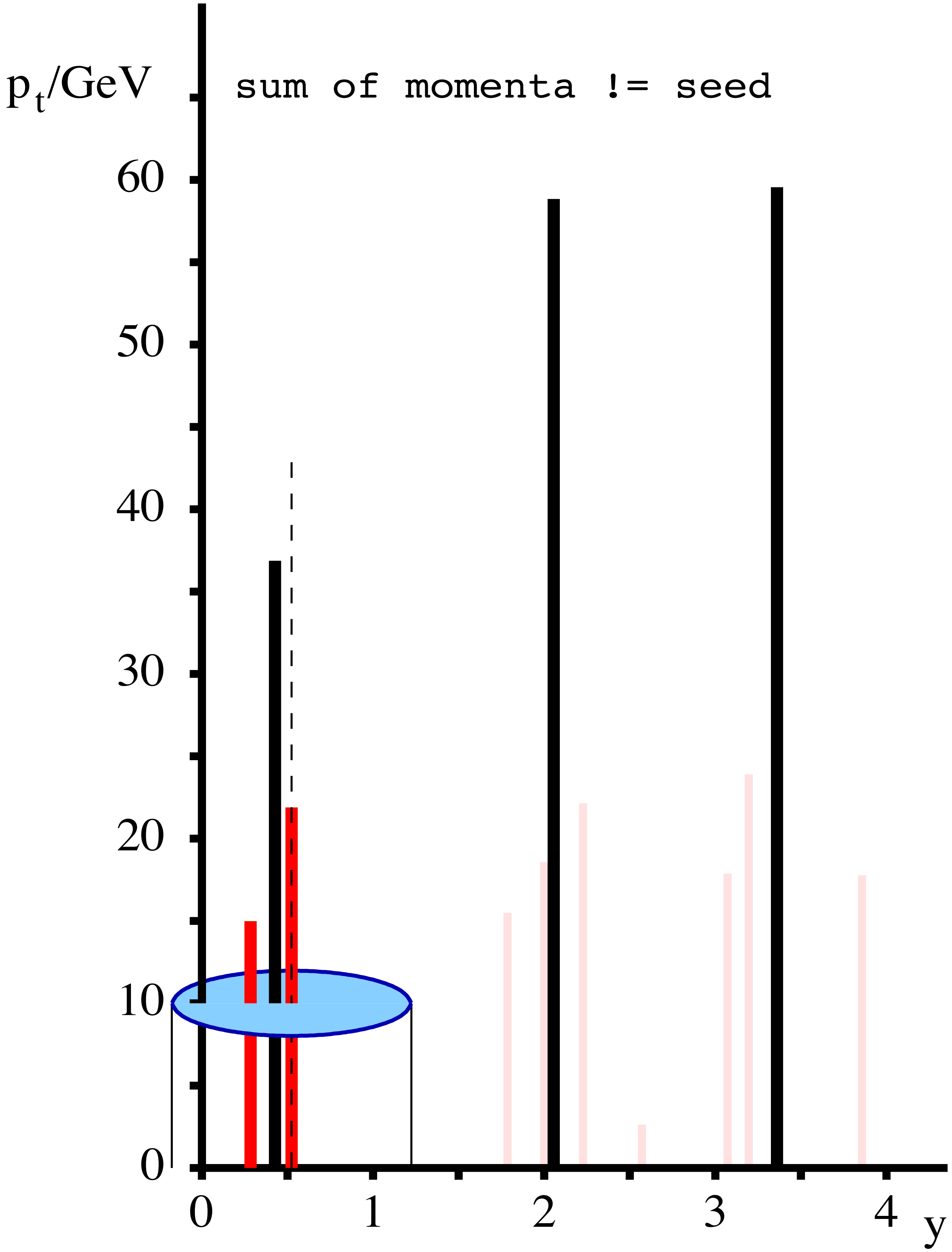
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

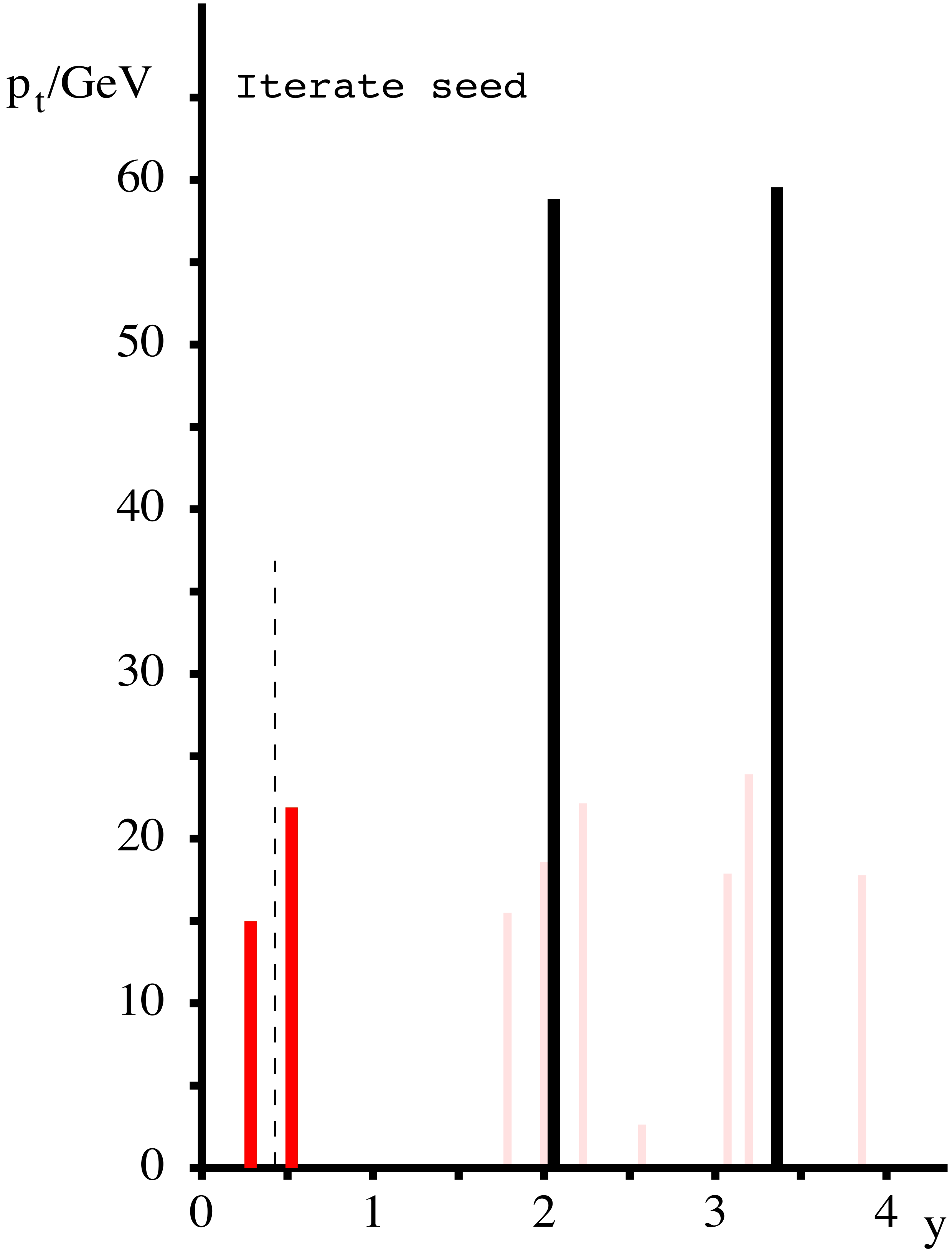
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

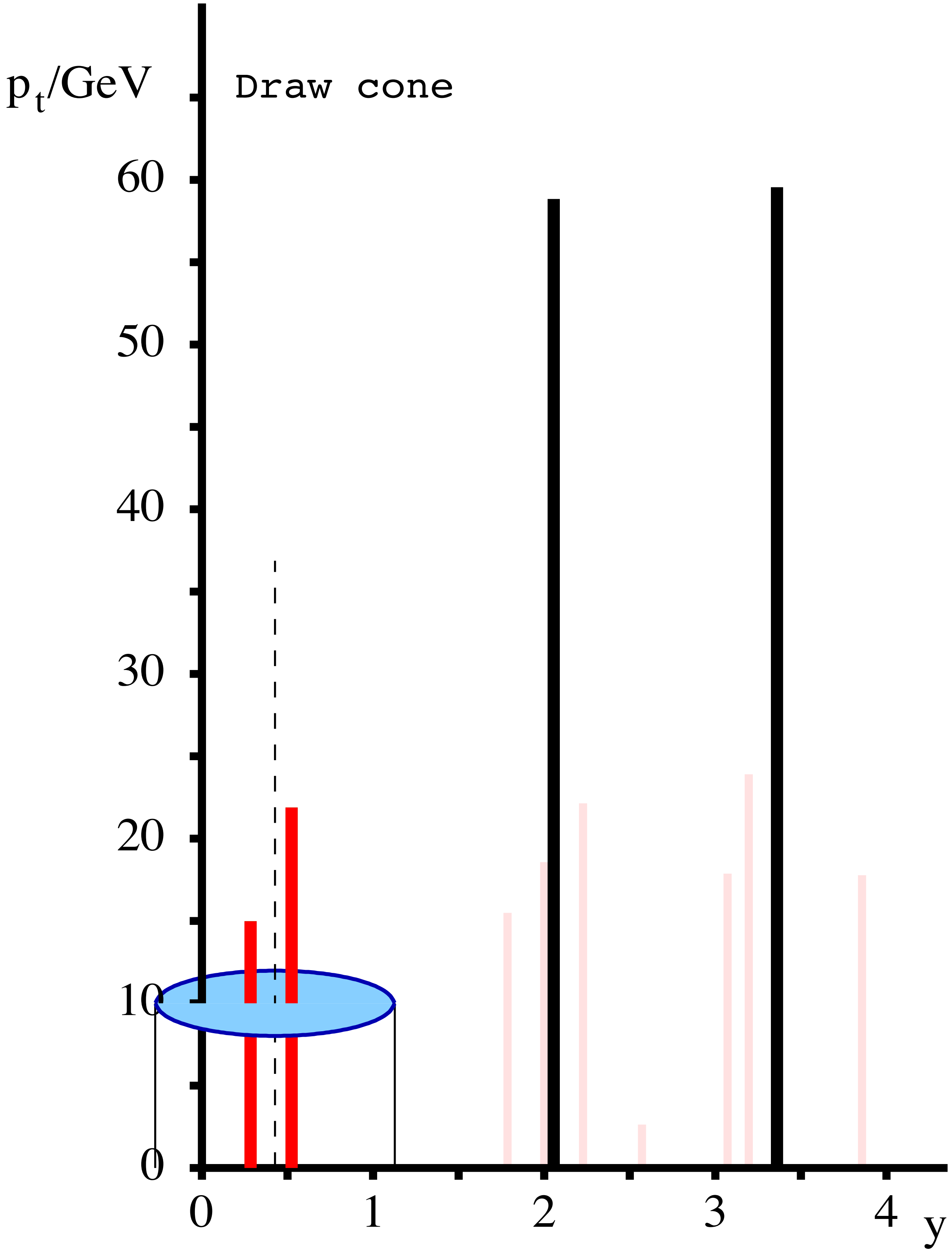
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

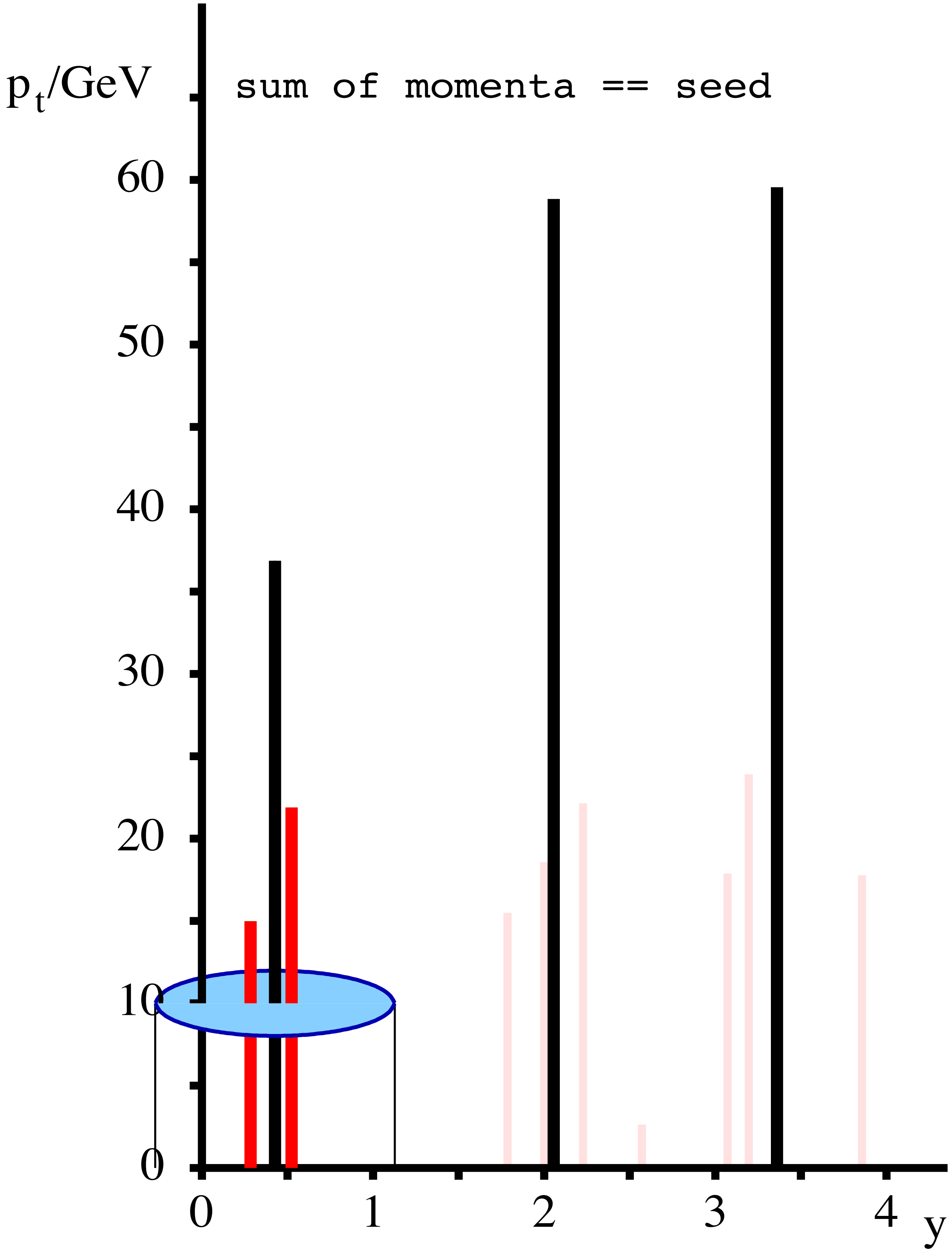
- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event



One of the simplest of the cone algs

e.g. CMS iterative cone

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



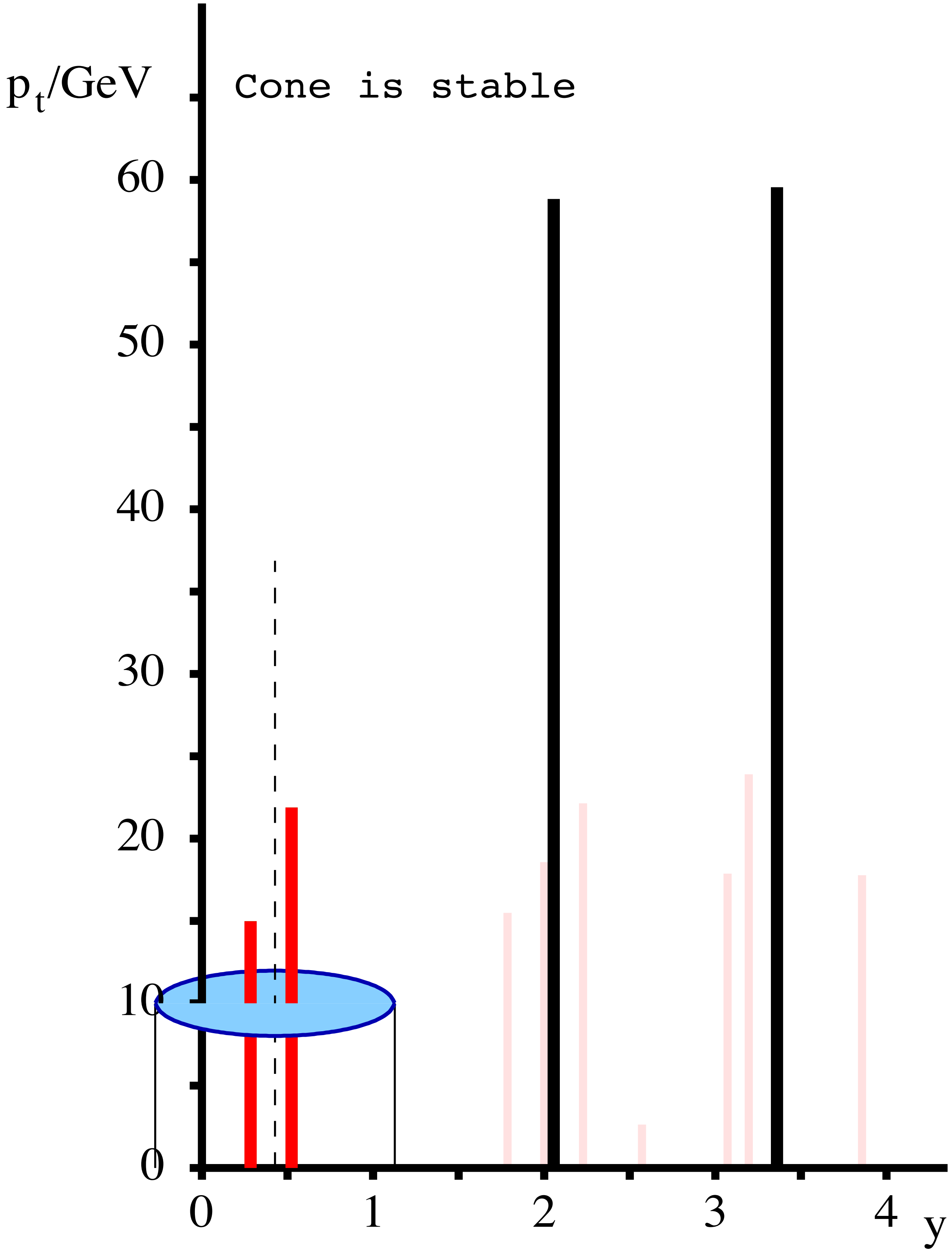
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

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



One of the simplest of the cone algs

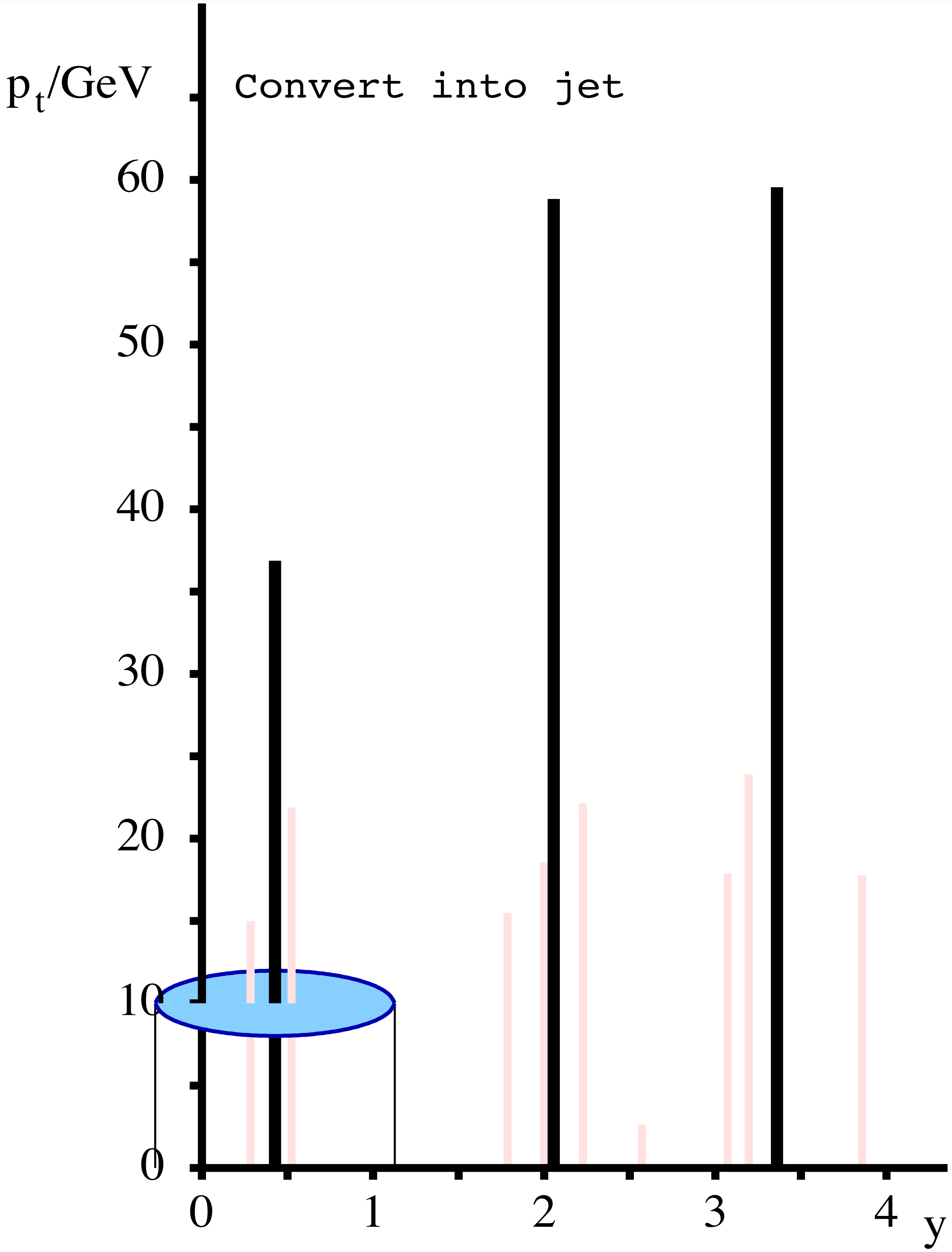
e.g. CMS iterative cone

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

Notes

- "Hardest particle" is collinear unsafe

more right away...



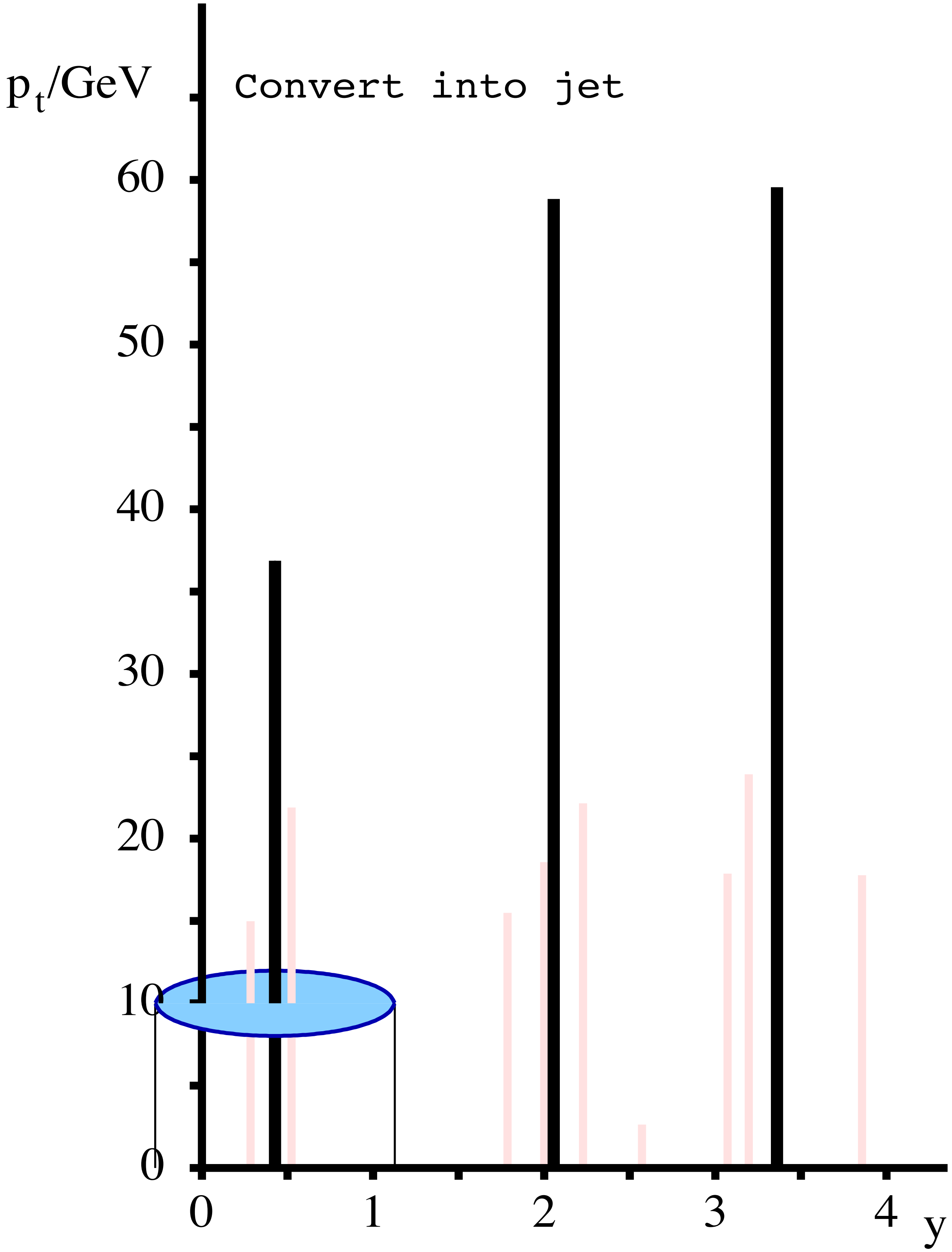
One of the simplest of the cone algs

e.g. CMS iterative cone

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

Notes

- "Hardest particle" is collinear unsafe more right away...



One of the simplest of the cone algs

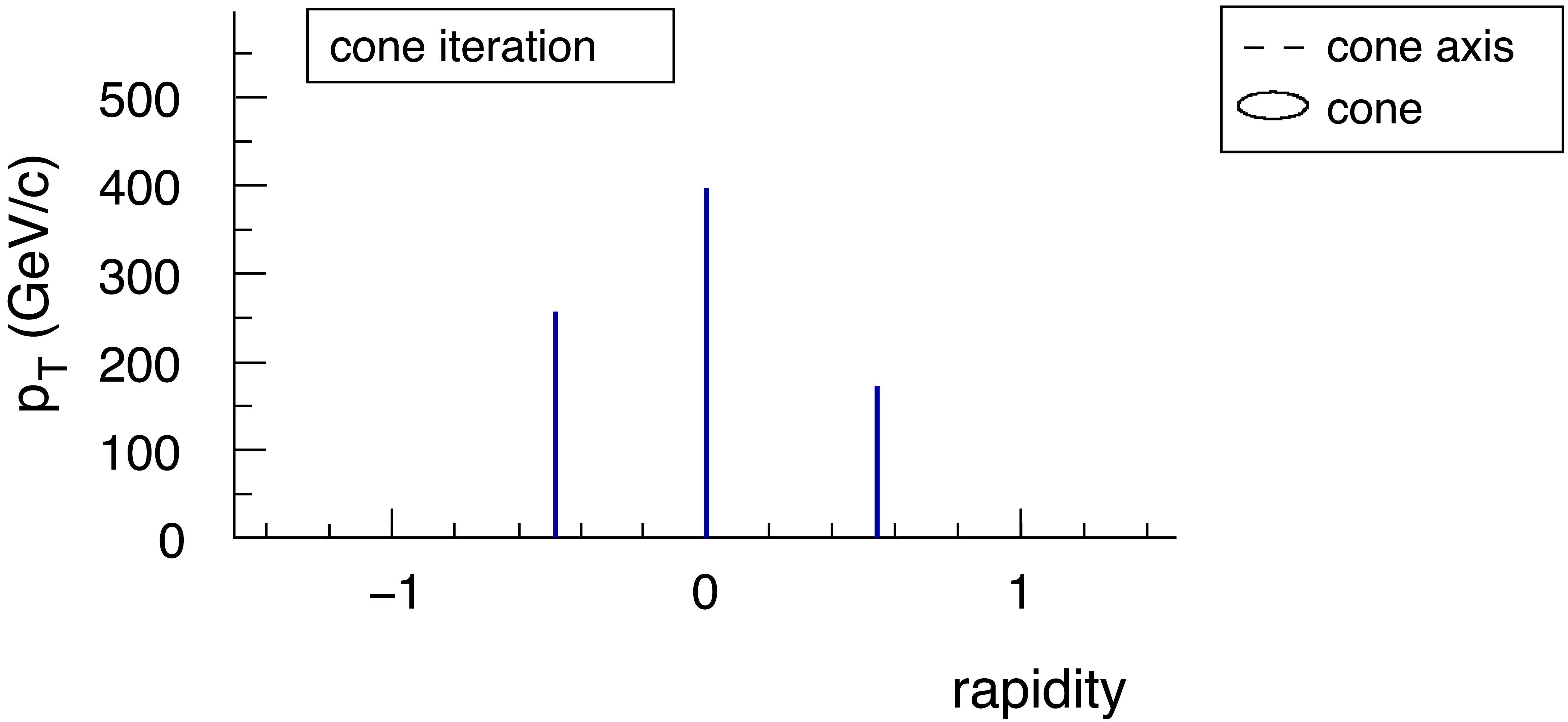
e.g. CMS iterative cone

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

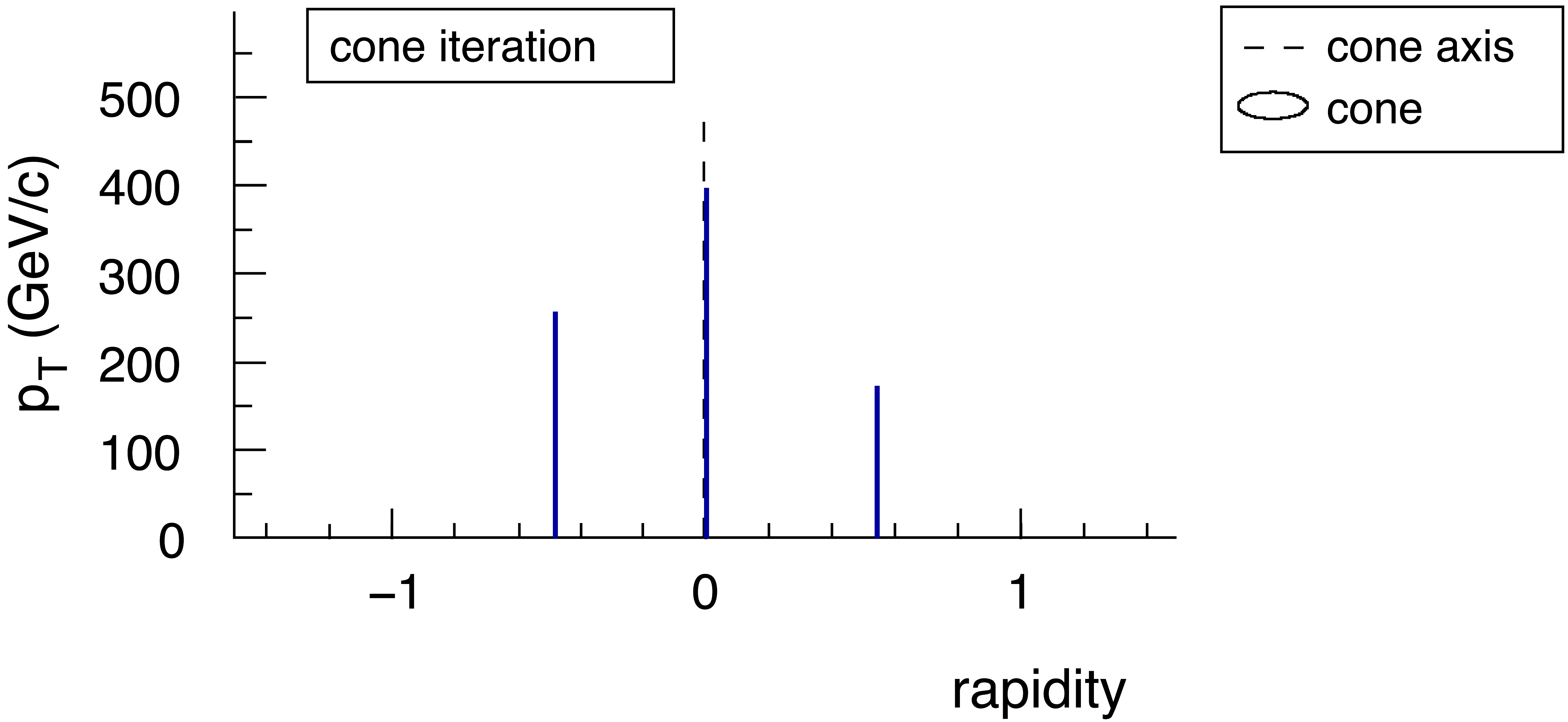
Notes

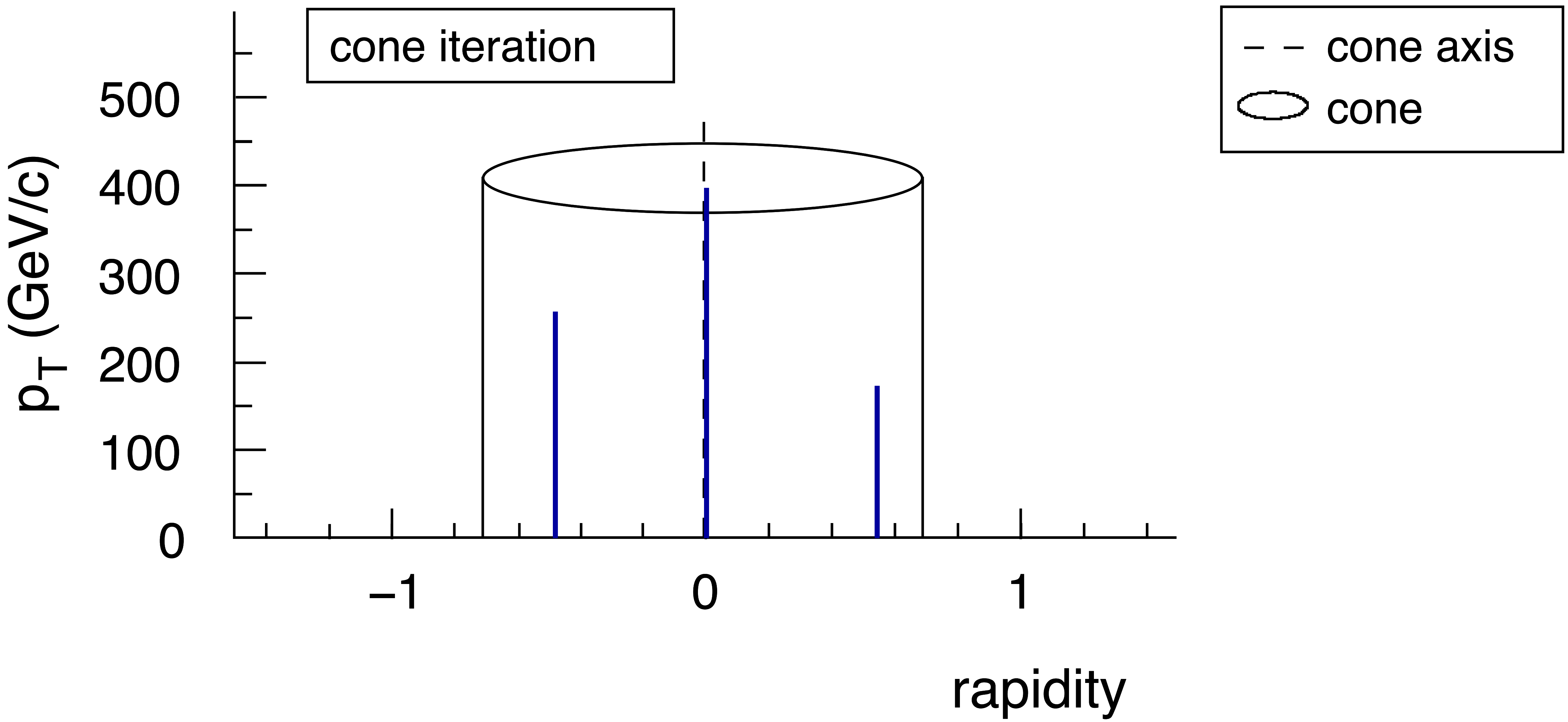
- "Hardest particle" is collinear unsafe

more right away...

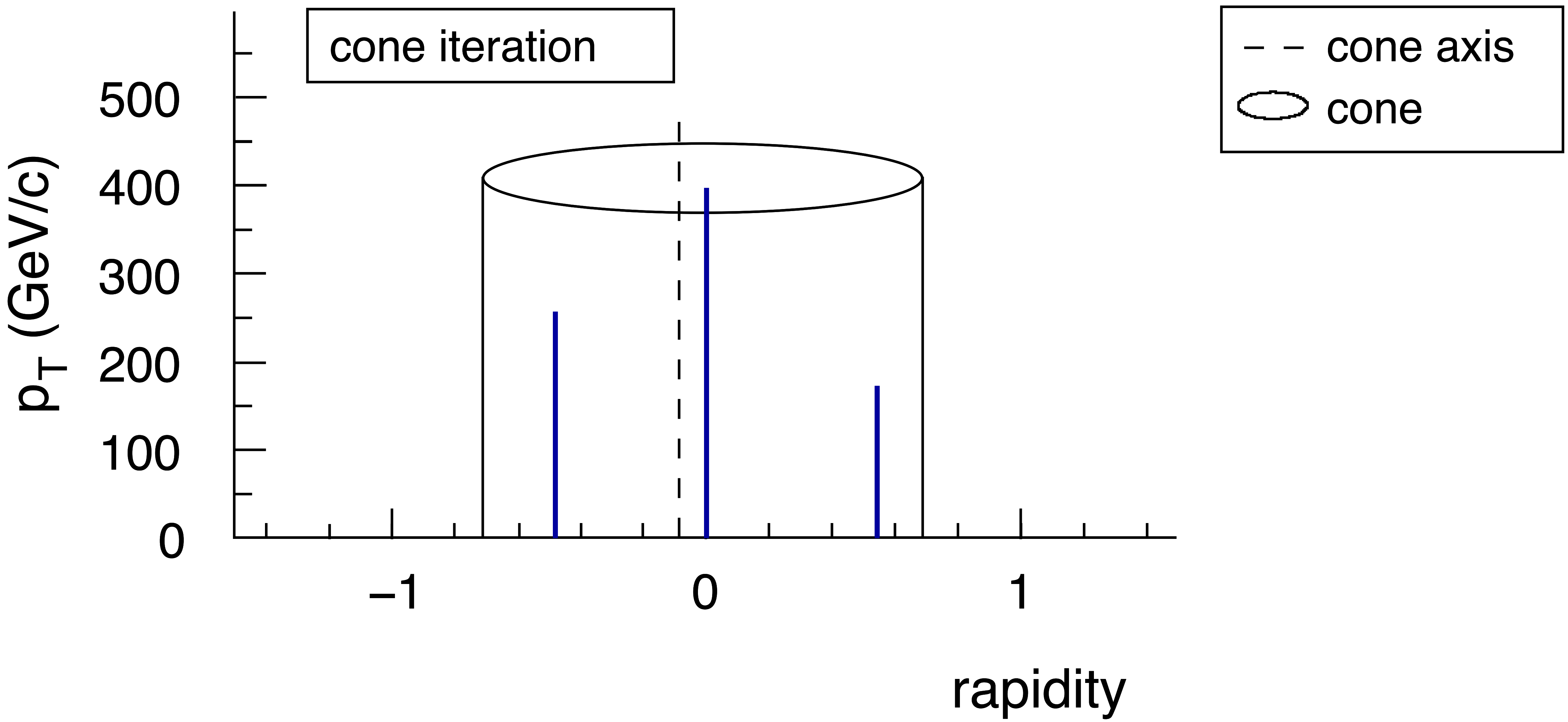

cone iteration
cone axis
cone
p_T (GeV/c)
500
400
300
200
100
0
−1
0
1
rapidity

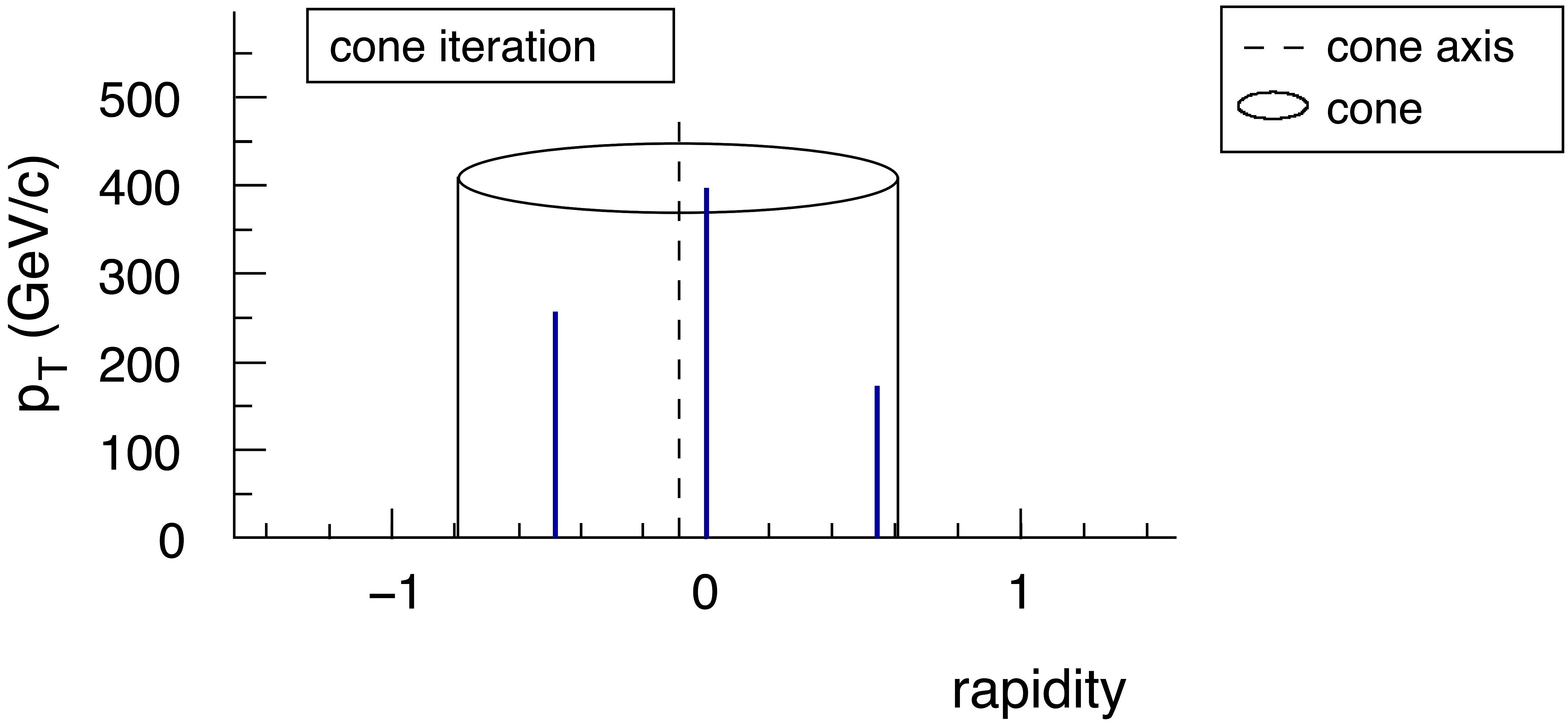
iterative cone issue



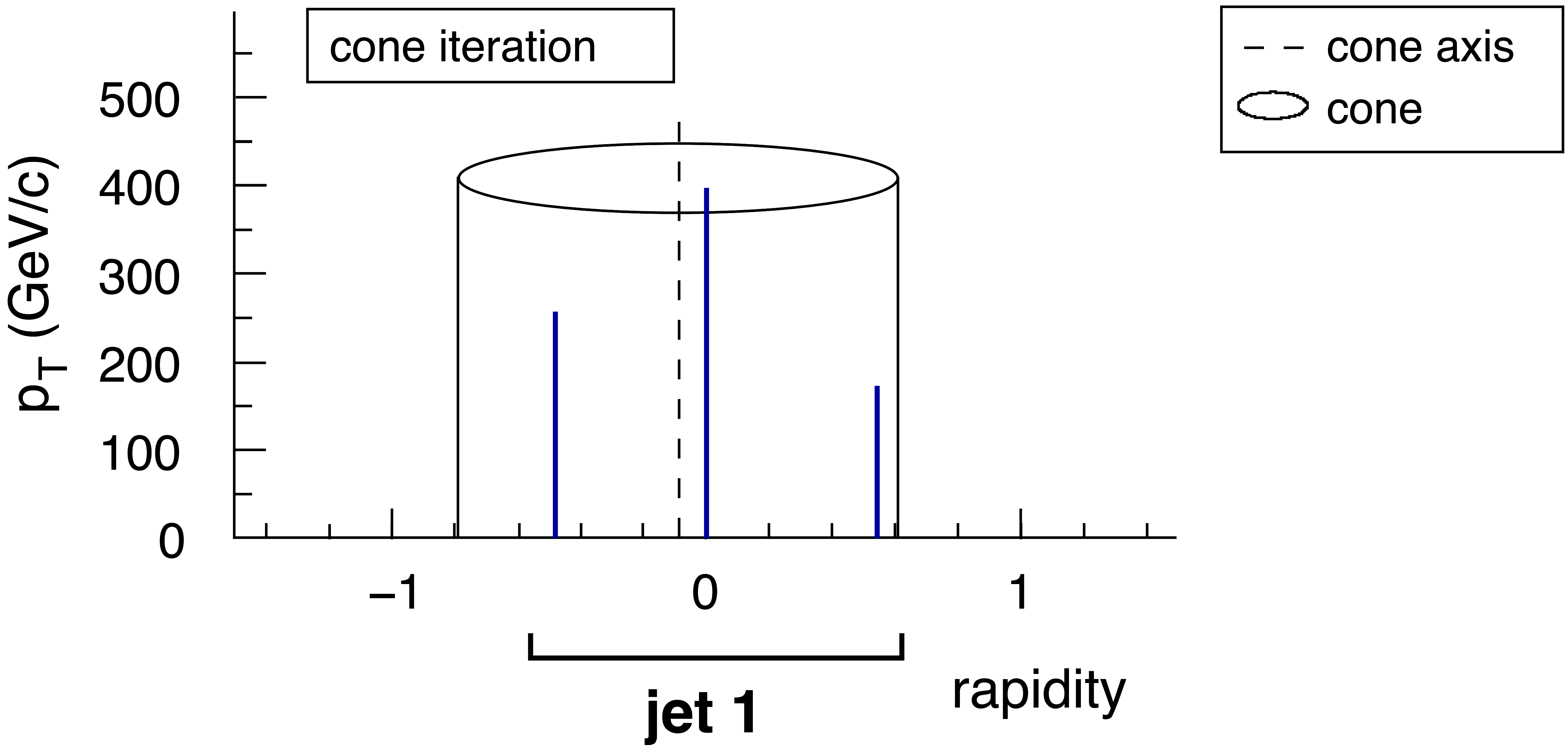

cone iteration
cone axis
cone
500
400
300
200
100
0
p_T (GeV/c)
−1
0
1
rapidity

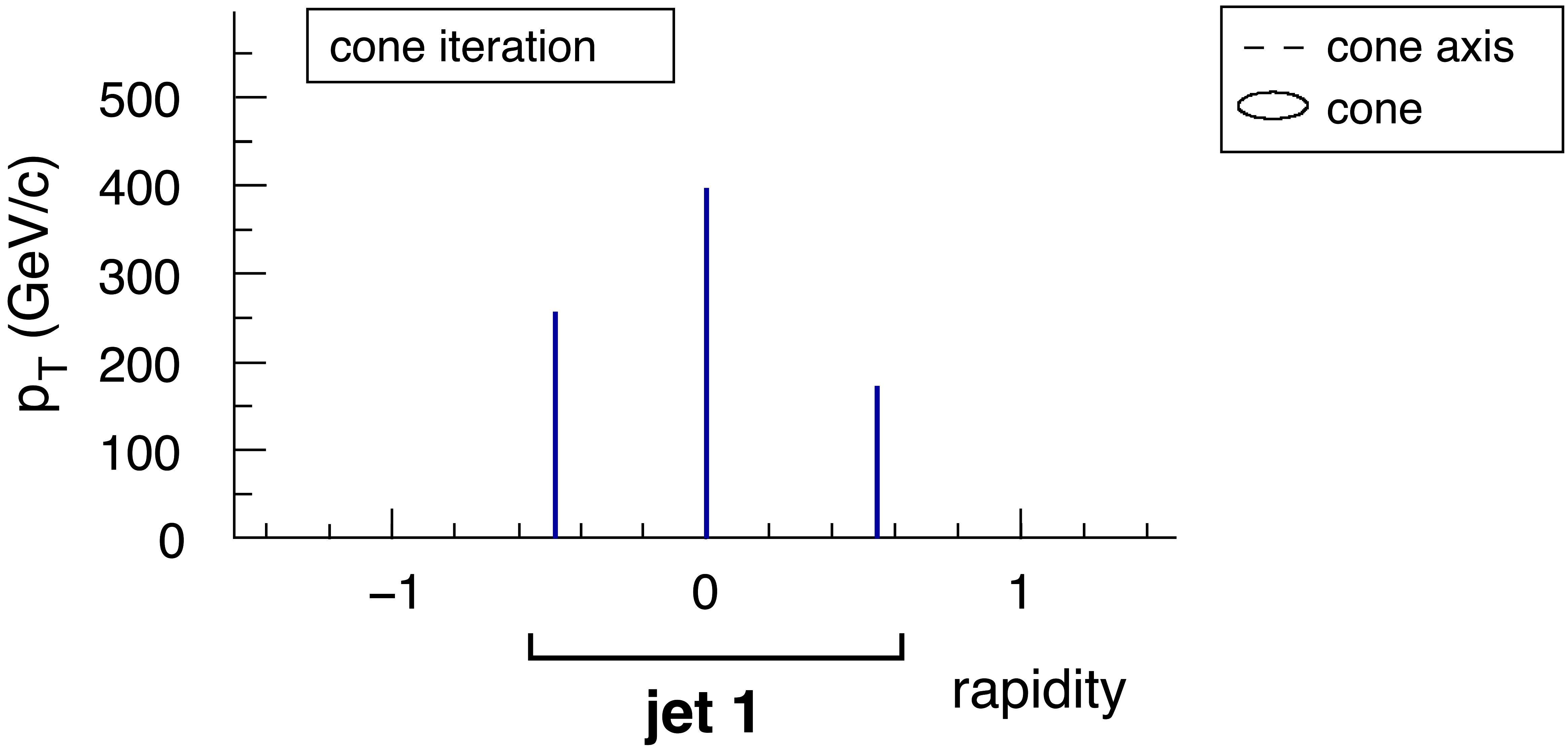
iterative cone issue

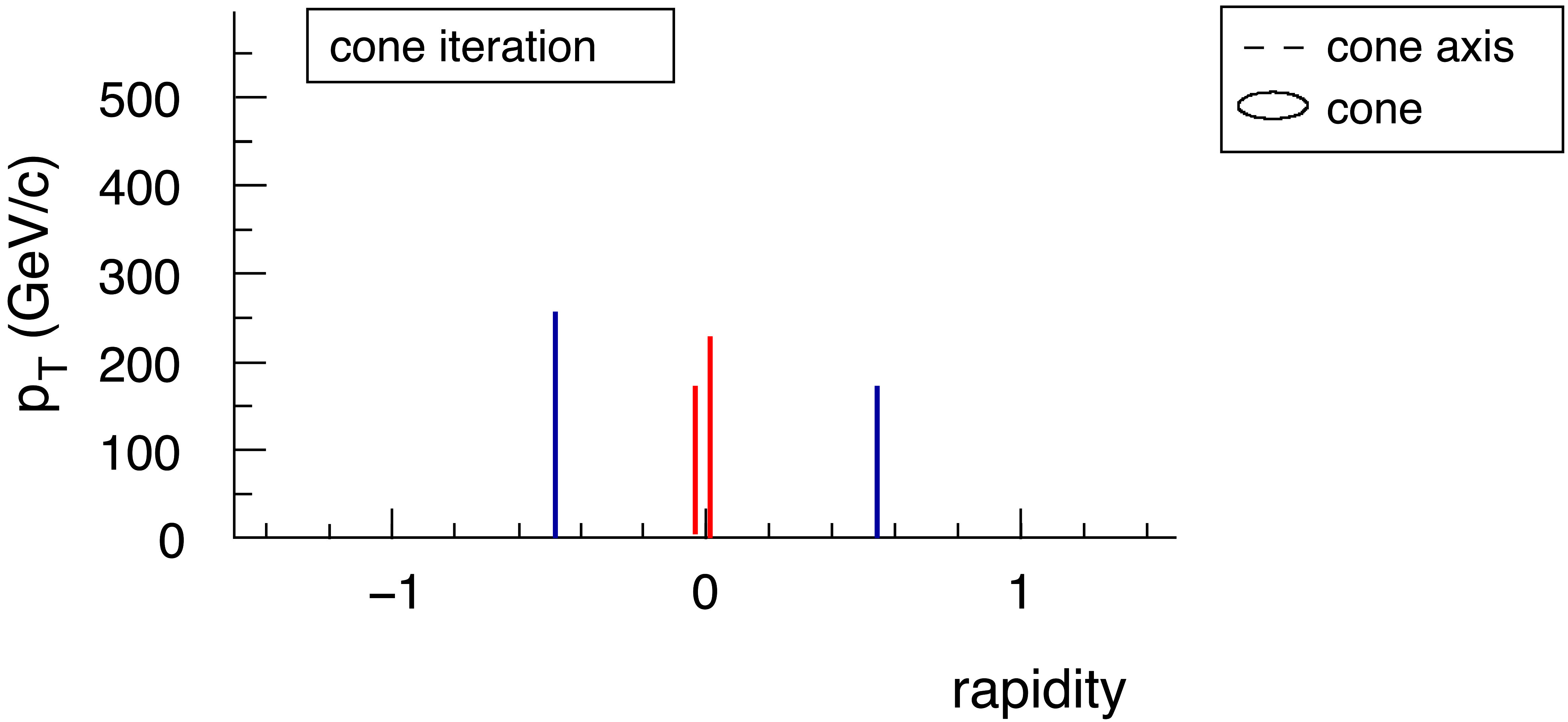



cone iteration
cone axis
cone
p_T (GeV/c)
500
400
300
200
100
0
–1
0
1
rapidity

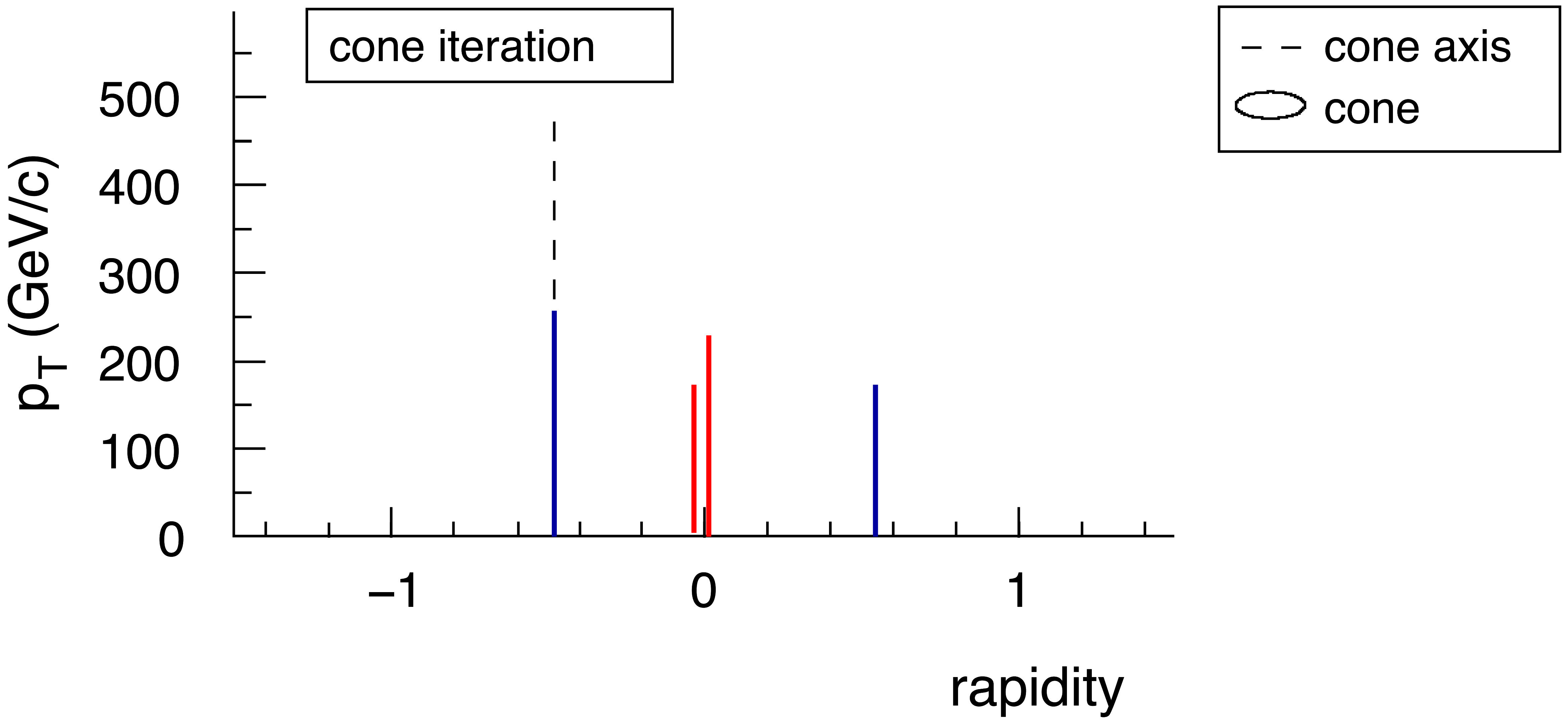
iterative cone issue

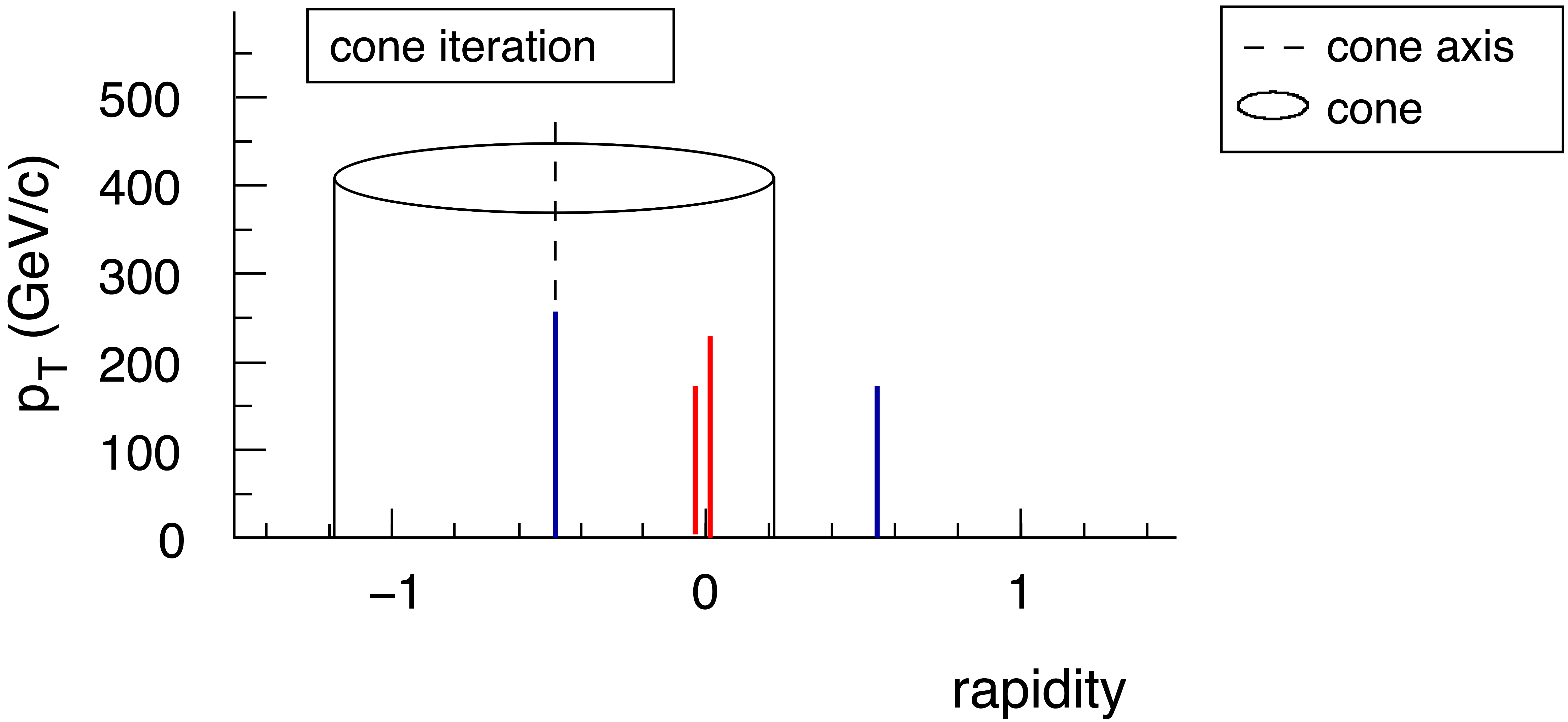


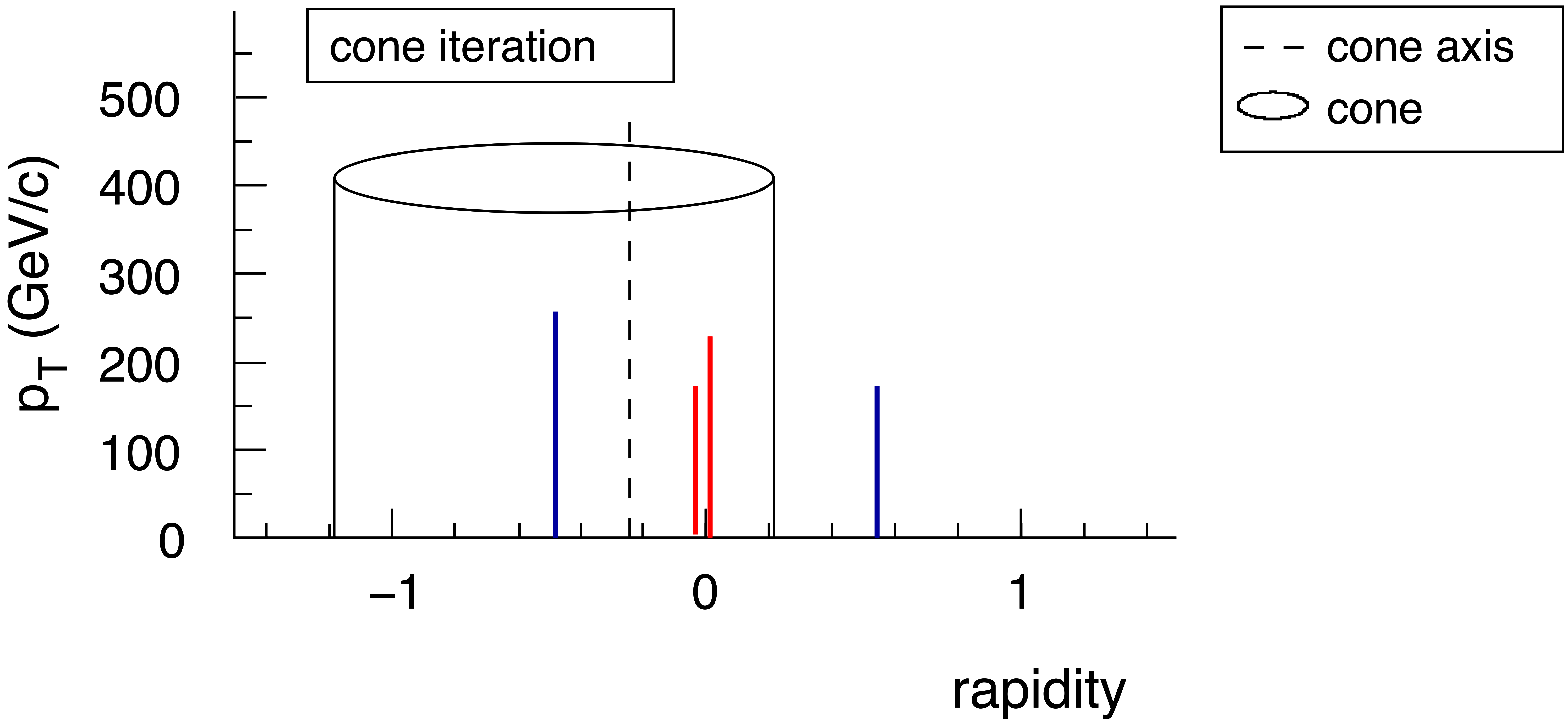

cone iteration
cone axis
cone
500
400
300
200
100
0
p_T (GeV/c)
–1
0
1
jet 1
rapidity


cone iteration
cone axis
cone
500
400
300
200
100
0
p_T (GeV/c)
–1
0
1
rapidity

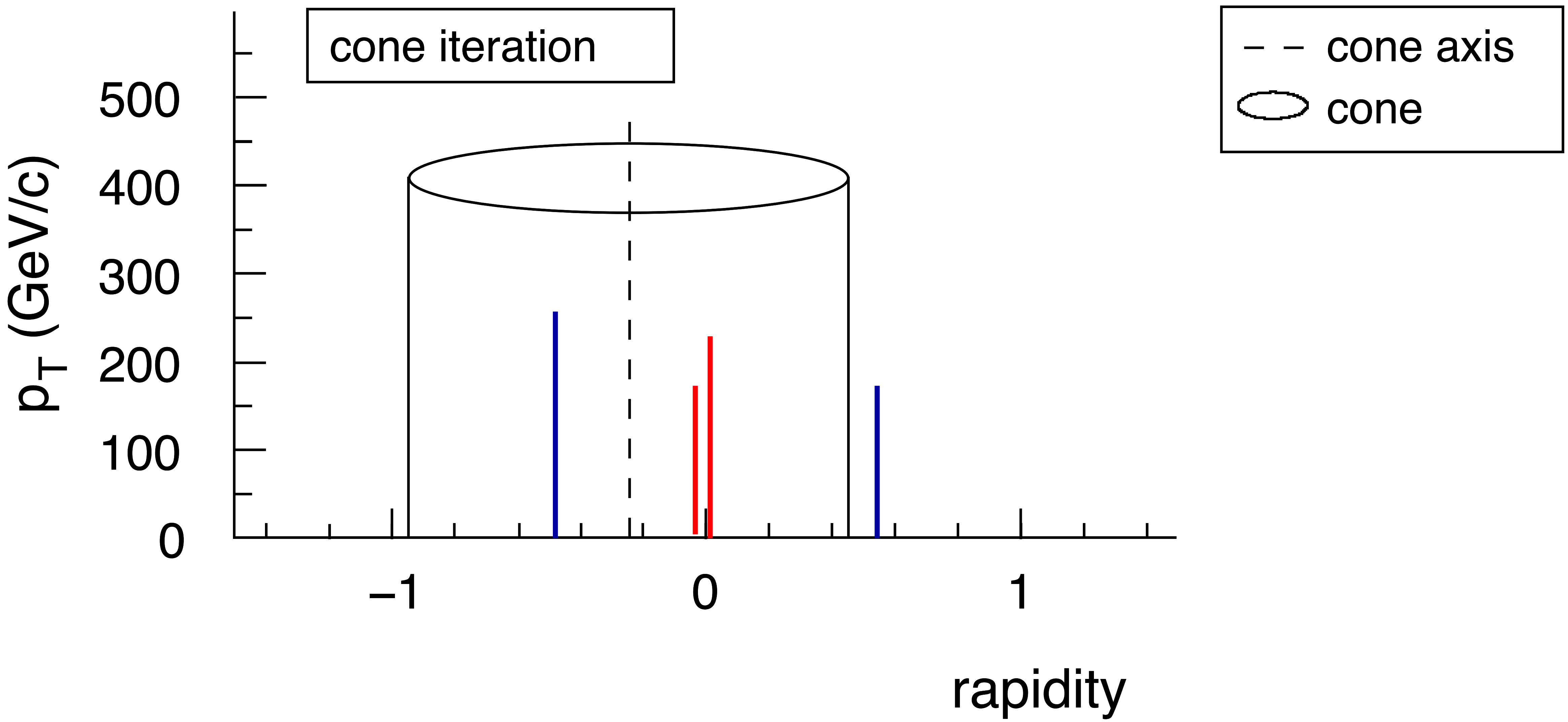
iterative cone issue



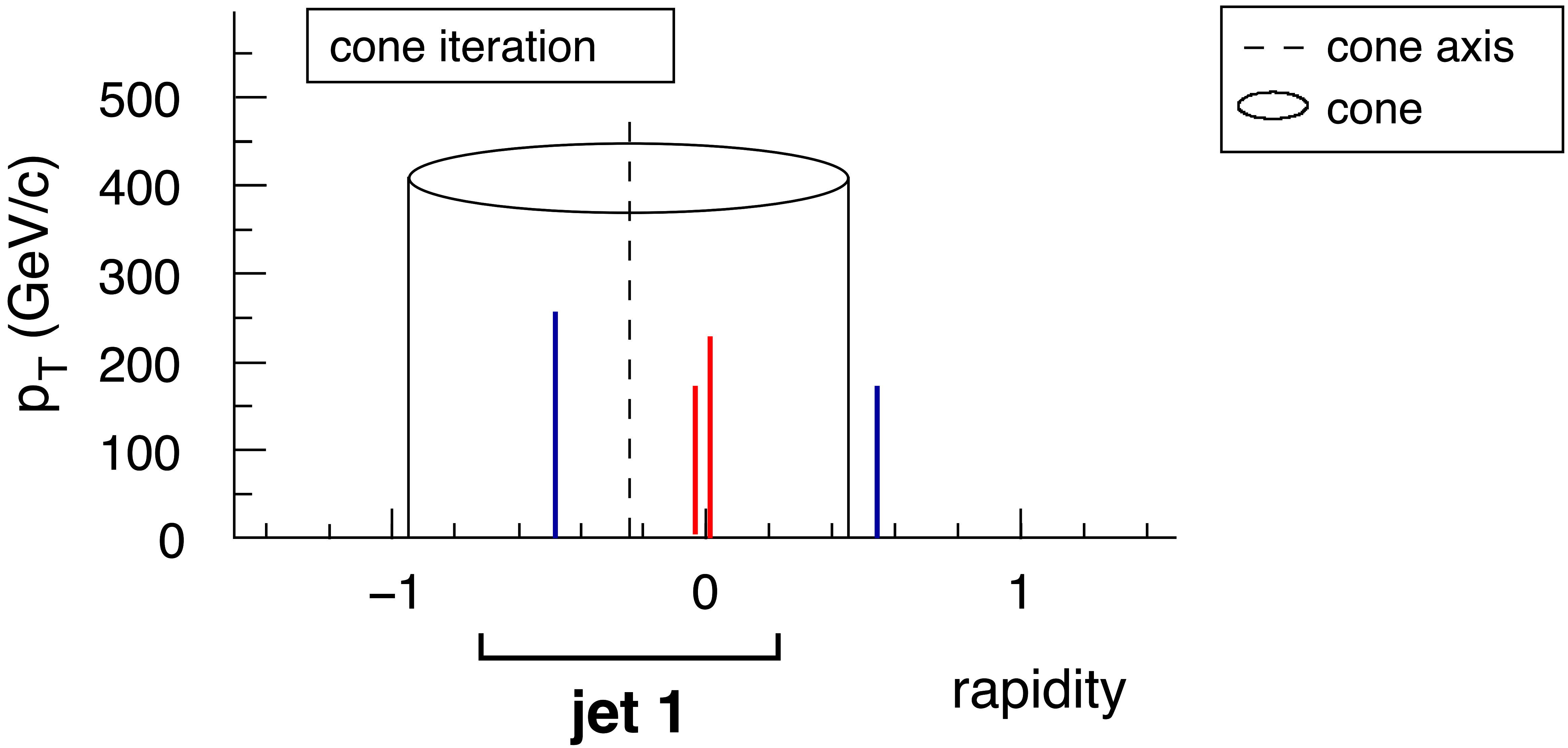

cone iteration
cone axis
cone
500
400
300
200
100
0
p_T (GeV/c)
−1
0
1
rapidity

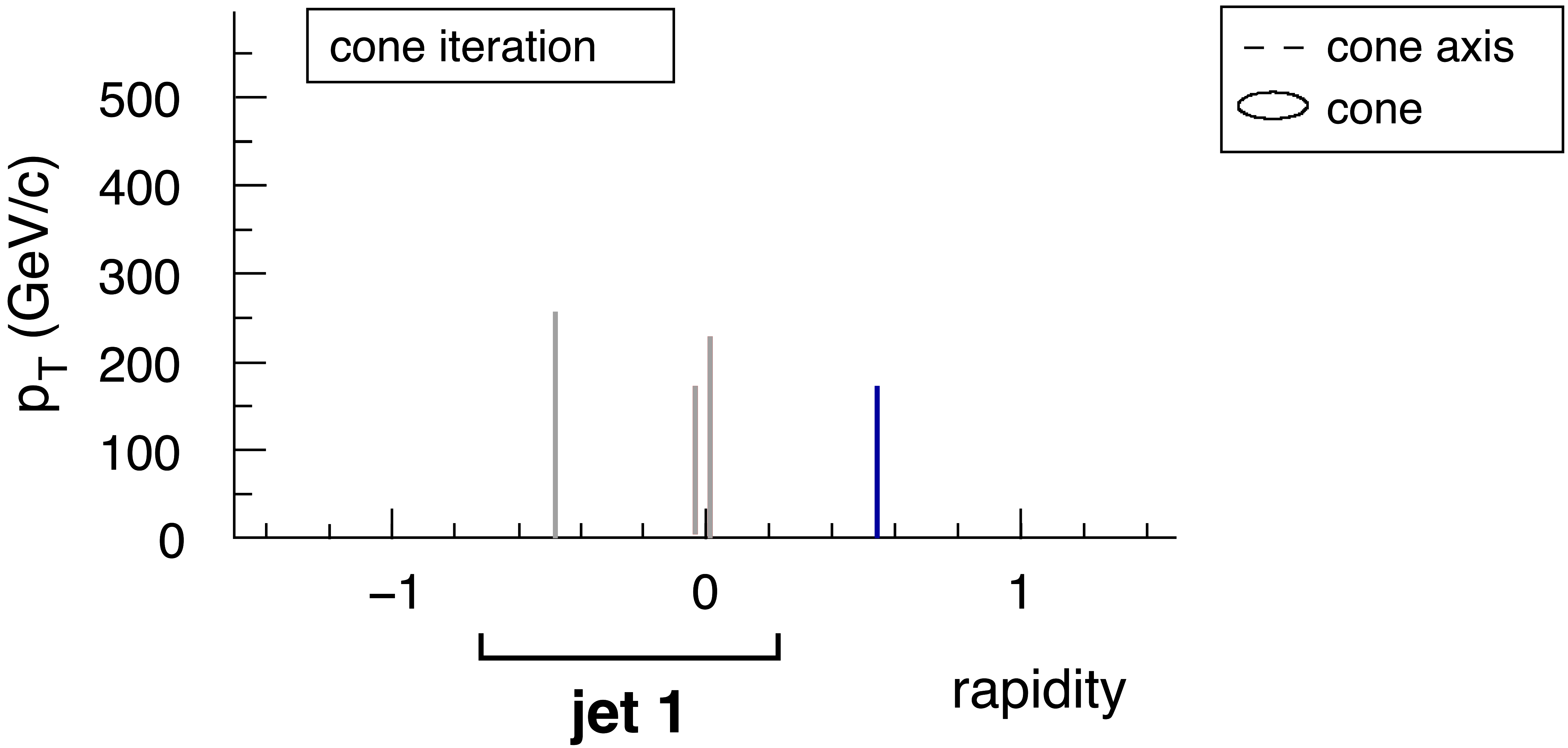

cone iteration
cone axis
cone
500
400
300
200
100
0
p_T (GeV/c)
–1
0
1
rapidity

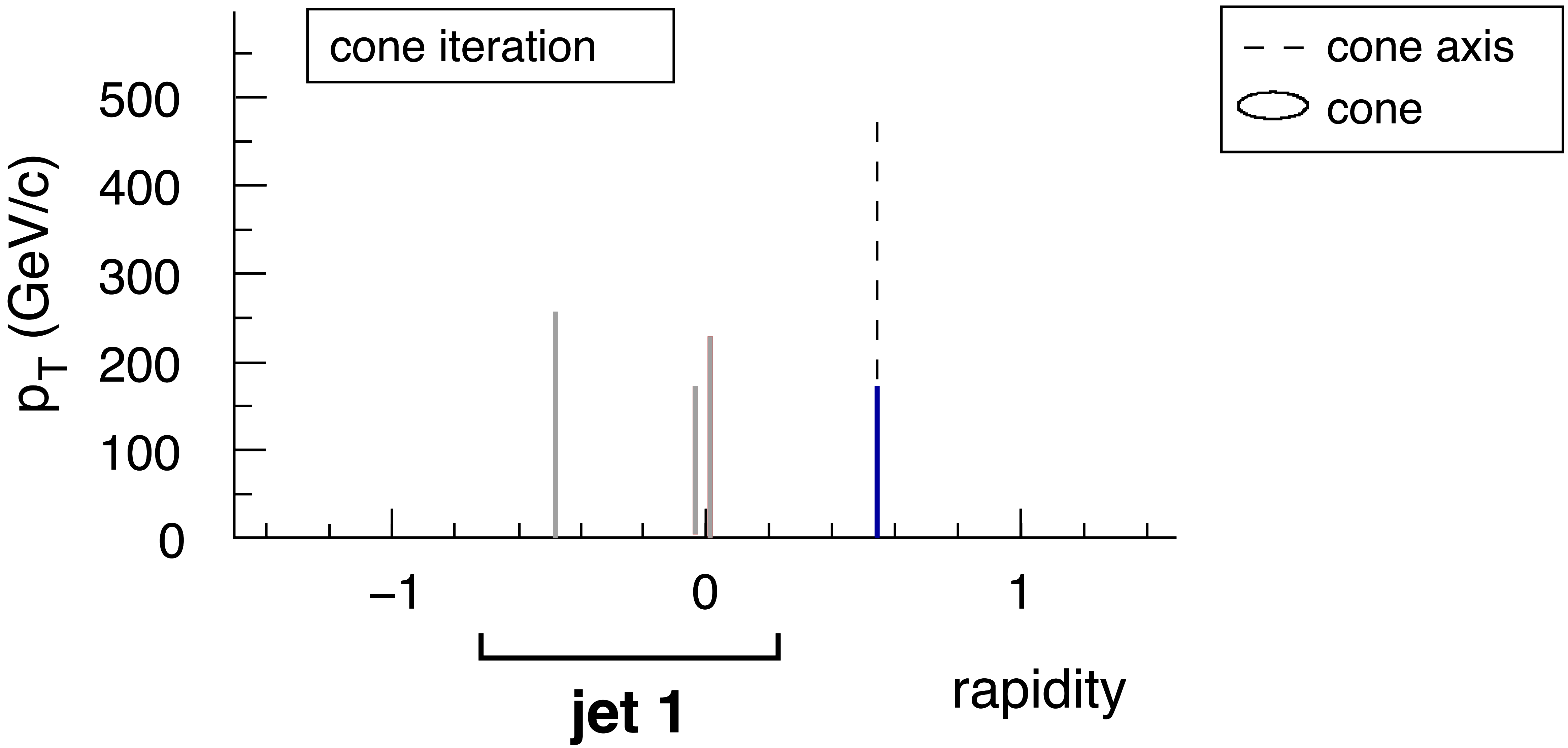
iterative cone issue



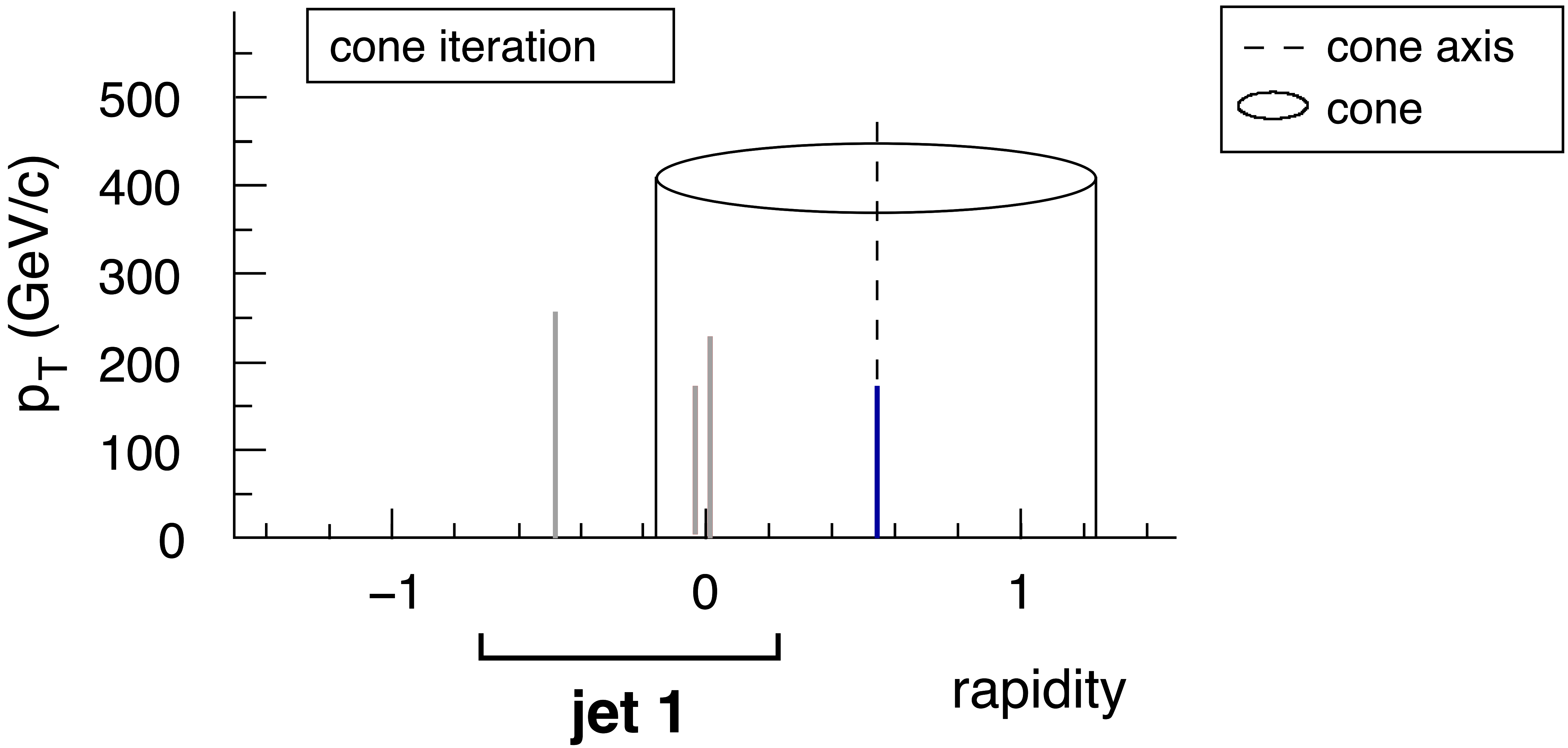
iterative cone issue

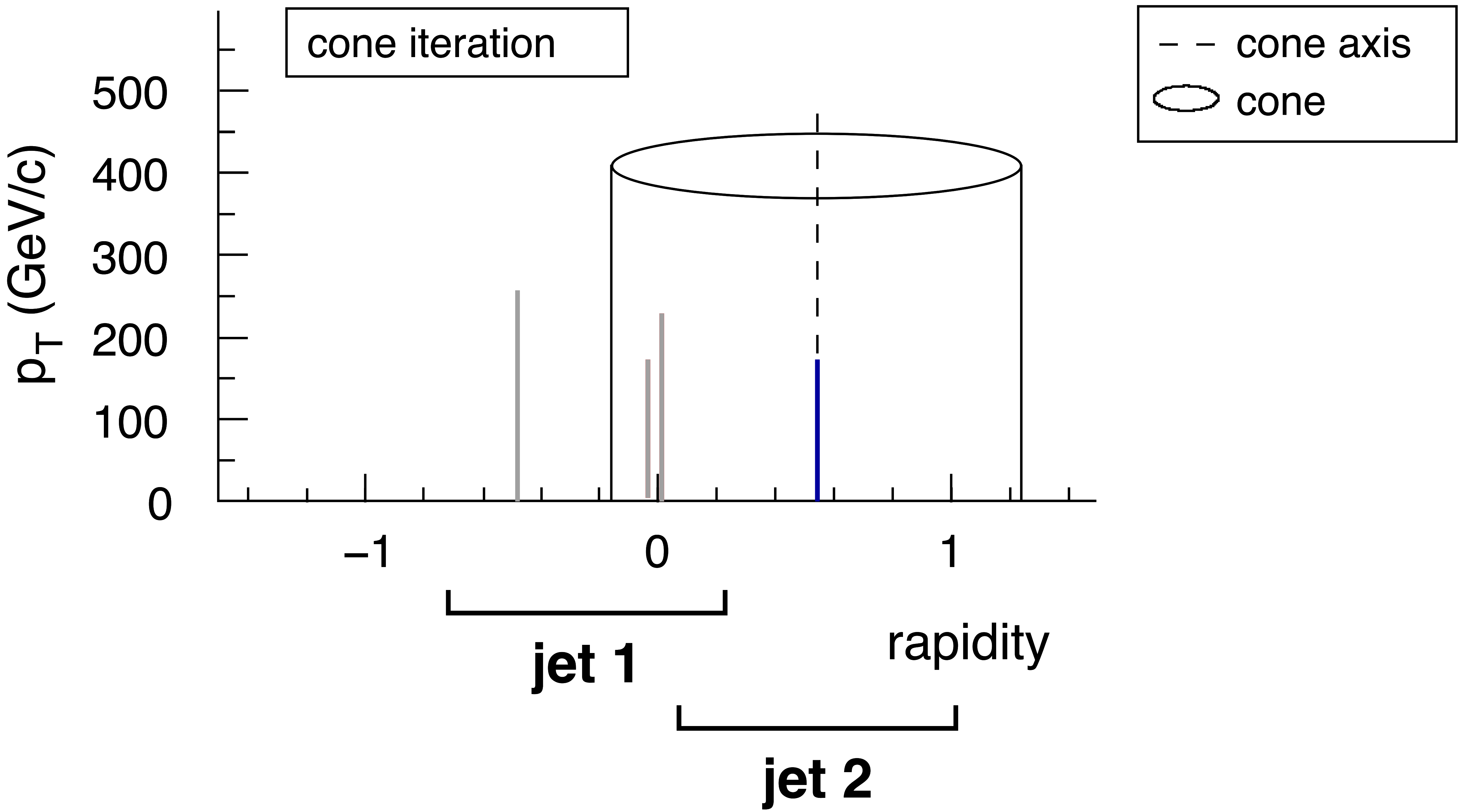


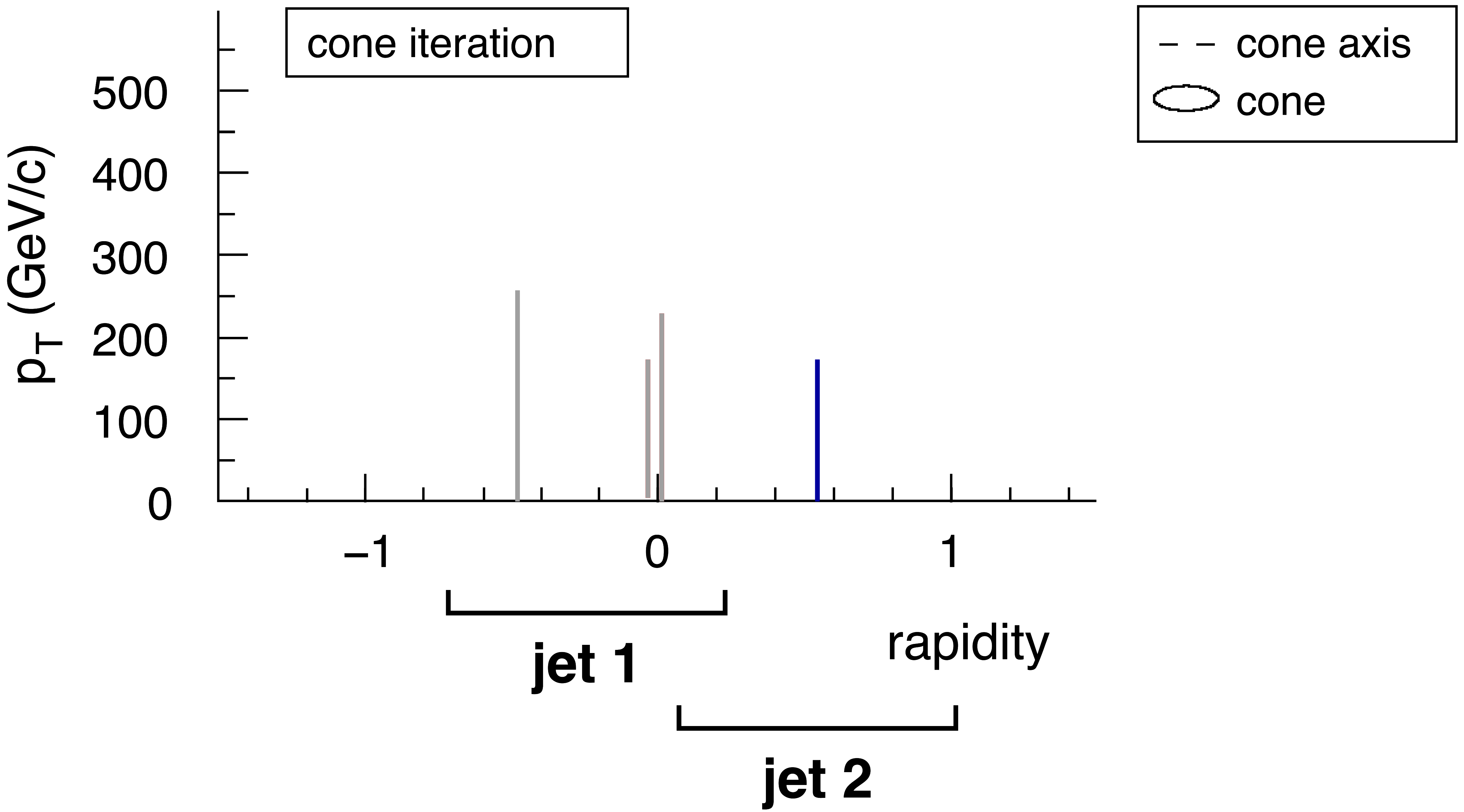

cone iteration
cone axis
cone
500
400
300
200
100
0
p_T (GeV/c)
–1
0
1
jet 1
rapidity


cone iteration
cone axis
cone
500
400
300
200
100
0
p_T (GeV/c)
−1
0
1
jet 1
rapidity

iterative cone issue

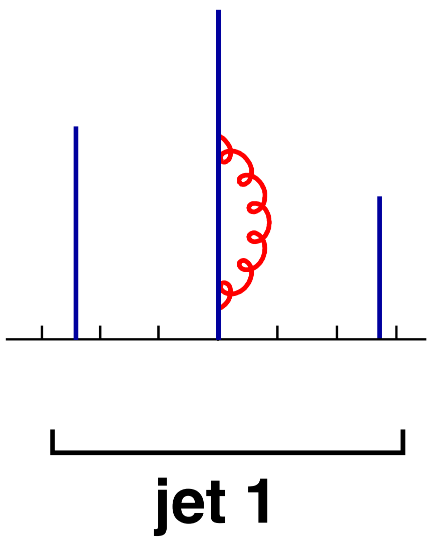



cone iteration
cone axis
cone
p_T (GeV/c)
500
400
300
200
100
0
−1
0
1
rapidity
jet 1
jet 2

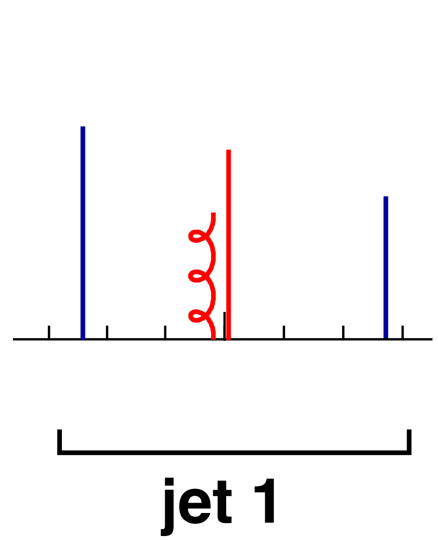


Collinear splitting can modify the final hard jets
The algorithm is **collinear unsafe**

Collinear Safe



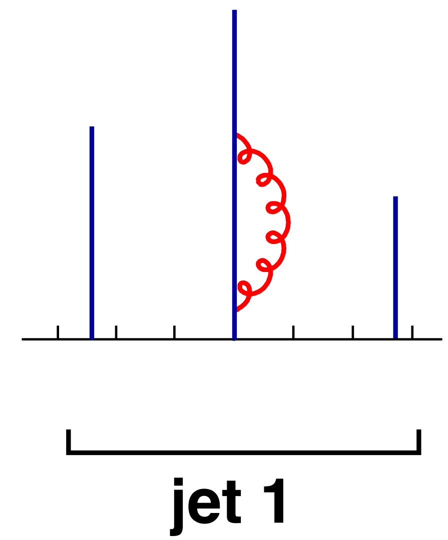
$\alpha_s^n \times (-\infty)$



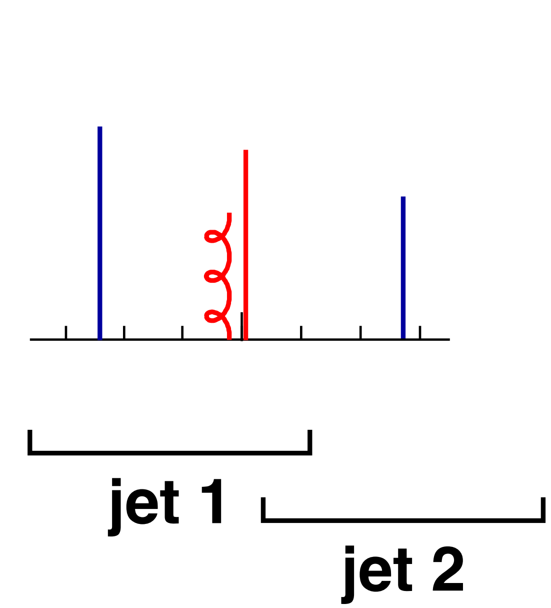
$\alpha_s^n \times (+\infty)$

Infinities cancel

Collinear Unsafe



$\alpha_s^n \times (-\infty)$



$\alpha_s^n \times (+\infty)$

Infinities do not cancel

Invalidates perturbation theory

The full cross section that you measure in experiment should correspond to an expression looking roughly as follows:

$$\sigma^{full} = \sigma_{LO}\left(1 + \alpha_s c_1 + \alpha_s^2 c_2 + \alpha_s^3 c_3 + \ldots + \mathcal{O}\left(\frac{\Lambda_{QCD}}{p_t}\right)\right)$$

A perturbative series

plus a non-perturbative contribution, suppressed by a power of Λ_{QCD}/p_t

We don't have the technology to calculate the full series or the non-perturbative part. Typically, one might "just" calculate next-to-leading order

$$\sigma^{NLO} = \sigma_{LO}\left(1 + \alpha_s c_1\right)$$

The point to perturbation theory is that the $c_2\alpha_s^2$, etc. terms are small compared to the ones you have calculated — *hence (e.g.) NLO should be a good approximation.*

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

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

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 ($\to$ NNLO)
$W/Z + 1$ jet	LO	NLO	NLO	NLO ($\to$ NNLO)
3 jets	**none**	LO	LO	NLO [nlojet++]
$W/Z + 2$ jets	**none**	LO	LO	NLO [MCFM]
m_{jet} in $2j + X$	**none**	**none**	**none**	NLO [Blackhat/Rocket/...]

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

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

And LHC relies on QCD for background double-checks

extraction of cross sections, extraction of parameters

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

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

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 ($\to$ NNLO)
W/Z + 1 jet	LO	NLO	NLO	NLO ($\to$ NNLO)
3 jets	**none**	LO	LO	NLO [nlojet++]
W/Z + 2 jets	**none**	LO	LO	NLO [MCFM]
m_{jet} in $2j + X$	**none**	**none**	**none**	NLO [Blackhat/Rocket/...]

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

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

And LHC relies on QCD for background double-checks

extraction of cross sections, extraction of parameters

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

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

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 ($\to$ NNLO)
3 jets	**none**	LO	LO	NLO [nlojet++]
$W/Z + 2$ jets	**none**	LO	LO	NLO [MCFM]
m_{jet} in $2j + X$	**none**	**none**	**none**	NLO [Blackhat/Rocket/...]

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

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

And LHC relies on QCD for background double-checks

extraction of cross sections, extraction of parameters

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

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

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 ($\to$ NNLO)
3 jets	none	LO	LO	NLO [nlojet++]
$W/Z + 2$ jets	none	LO	LO	NLO [MCFM]
m_{jet} in $2j + X$	none	none	none	NLO [Blackhat/Rocket/...]

...stment in NLO

Multi-jet contexts much more sensitive: ubiquitous at LHC

And LHC relies on QCD for background double-checks

extraction of cross sections, extraction of parameters

There is a way of resolving this kind of problem for cone algorithms

SISCone

[Seedless Infrared Safe Cone]

But, in the end, this is not the solution that the LHC experiments chose, so let's spend our time on the one they did

Hierarchical clustering

[sequential recombination algorithms]

Sequential recombination algorithms (e^+e^-)

[aka hierarchical agglomerative clustering]

- Define a distance measure $\mathbf{d_{ij}}$ between all pairs of particles.
- Recombine pair with smallest d_{ij}
- Repeat until all $d_{ij} > d_{cut}$

The algorithm's general behaviour is governed by how one defines the interpair distances d_{ij}

d_{cut} is a **resolution parameter**, which governs how whether jets are coarse or fine objects

JADE algorithm – the original seq. rec. algorithm

$$d_{ij} = m_{ij}^2 \simeq 2E_iE_j(1-\cos\theta_{ij})$$

The most obvious choice?
But does not give sensible behaviour

[see blackboard]

Classic e^+e^- seq. rec. algorithms (2)

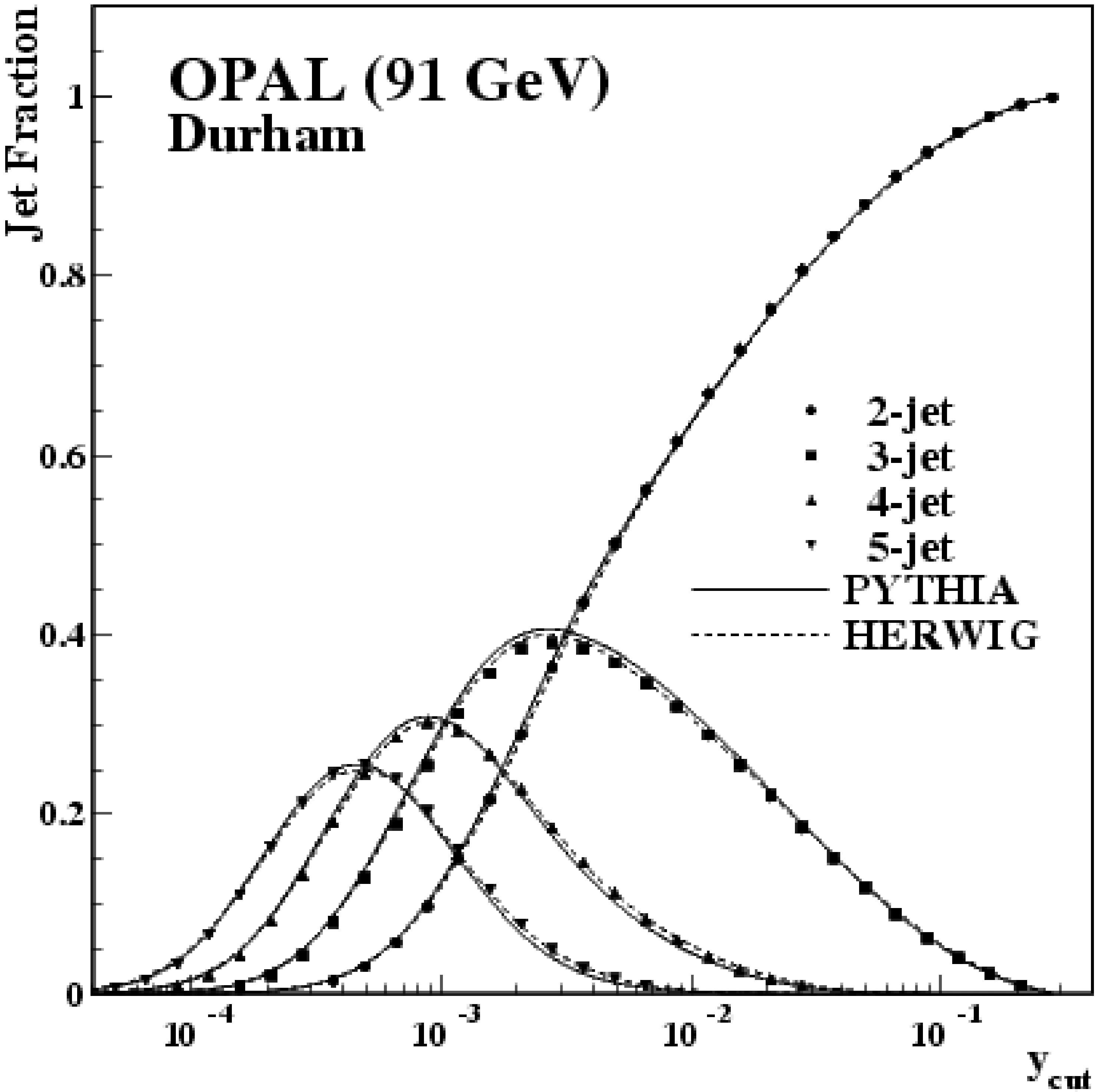
Use QCD divergences to help decide distance measure:

The stronger the divergence between a pair of particles, the more likely it is they should be associated with each other. *[see blackboard]*

k_t / Durham algorithm – the most widely used at LEP

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

Catani, Dokshitzer, Olsson, Turnock & Webber '91



Fraction of events with *n* jets, as a function of the resolution parameter

$$y_{cut} = d_{cut}/Q^2$$

What changes at a hadron collider?

- You have (unseen) beams, introduce a "beam distance"

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

squared trans. mom. wrt beam

- You want to use longitudinally invariant variables, i.e. p_t, rapidity (y) and azimuth (φ)

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

exclusive k_t algorithm

still just one parameter d_{cut}

Catani, Dokshitzer, Seymour & Webber 1993

Two parameters, *R* and $p_{t,min}$

(These are the two parameters in essentially every widely used hadron-collider jet algorithm)

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

Reformulate algorithm

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

   Only use jets with $p_t > p_{t,min}$

Inclusive k_t algorithm

S.D. Ellis & Soper, 1993

In what way do the inclusive and exclusive variants' behaviours differ?

*[see ~~black~~**white**board]*

Is one "right", the other wrong"?

A priori no, maybe we'll come back to this later.

k_t in action

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

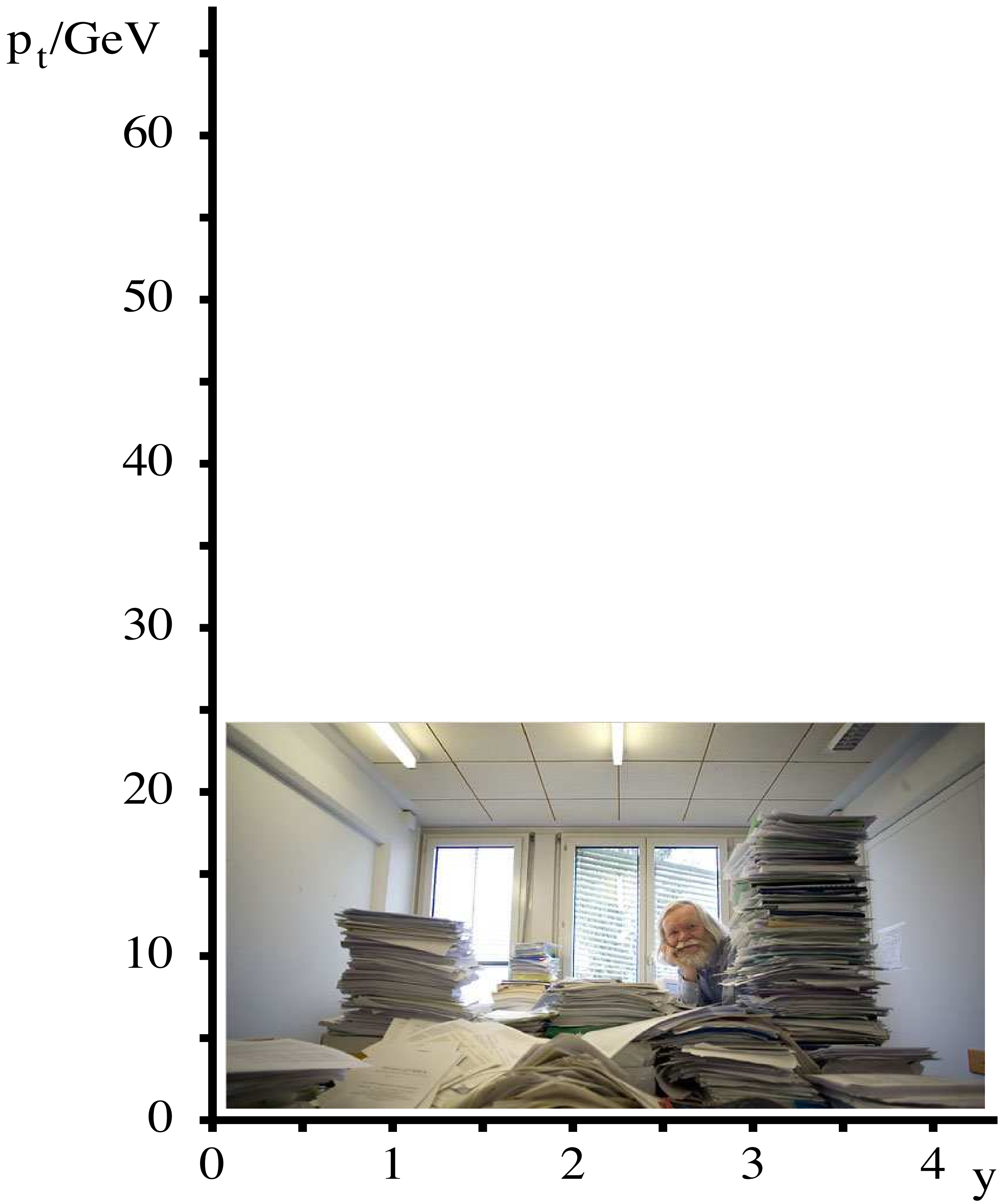
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

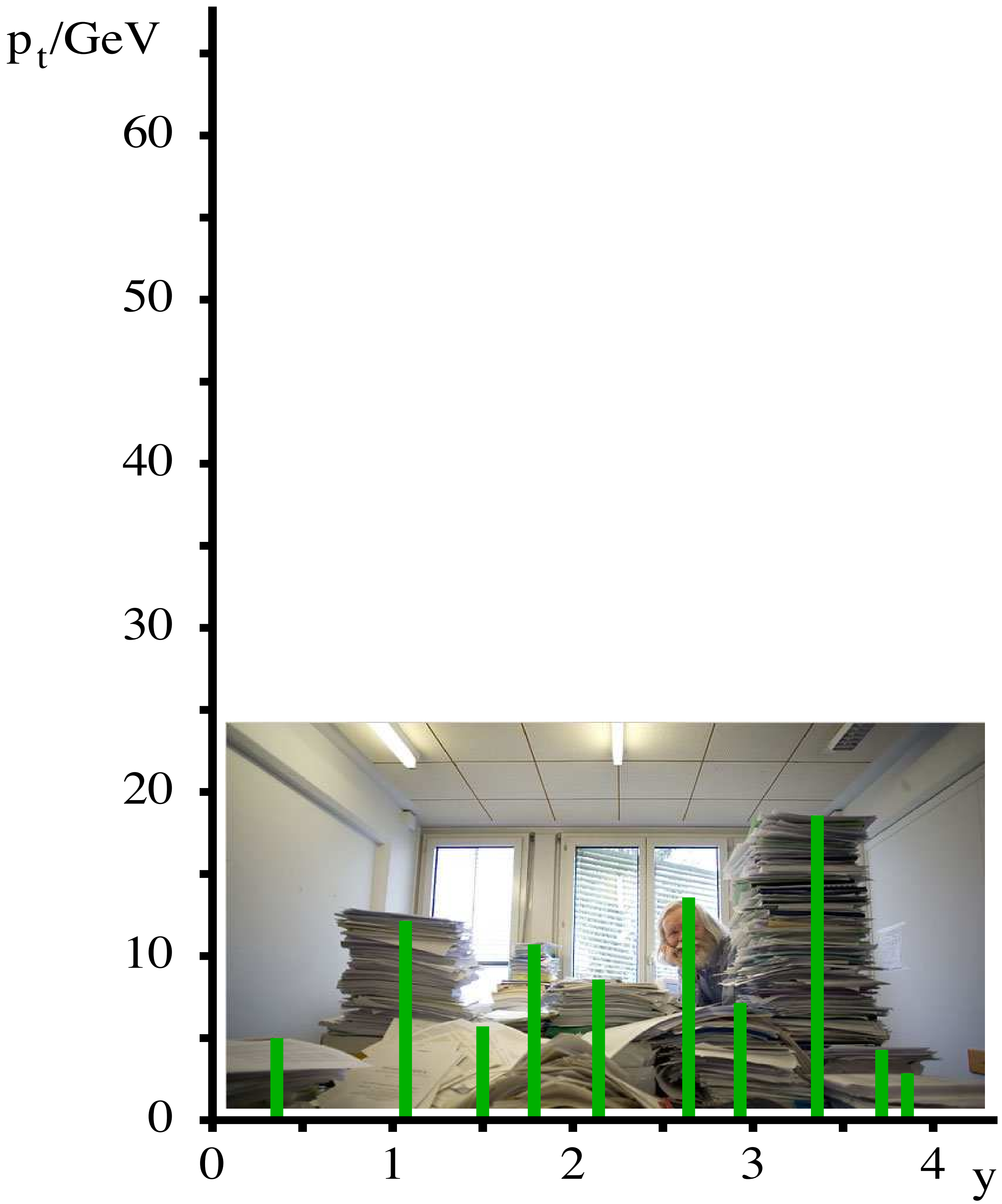
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

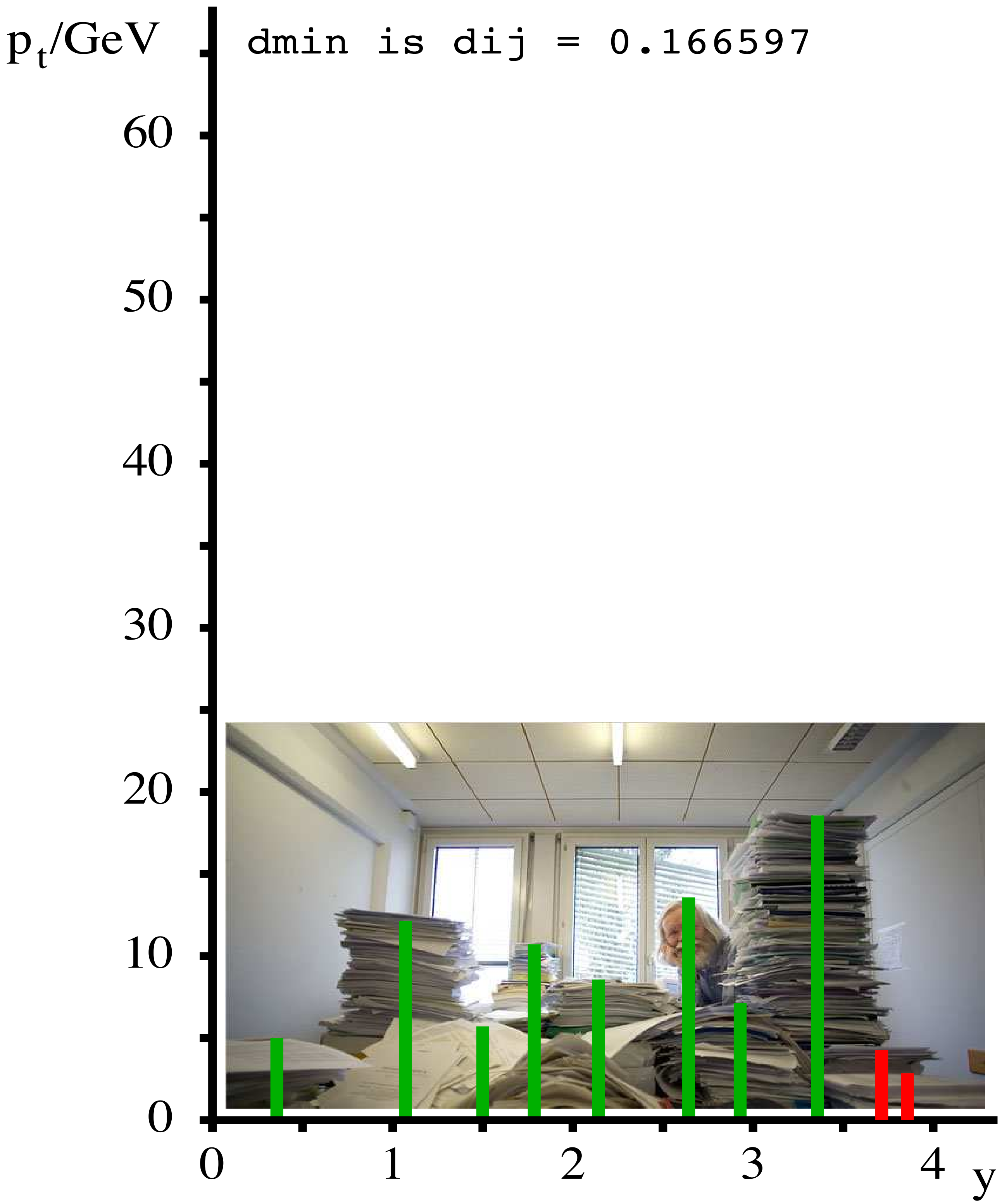
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

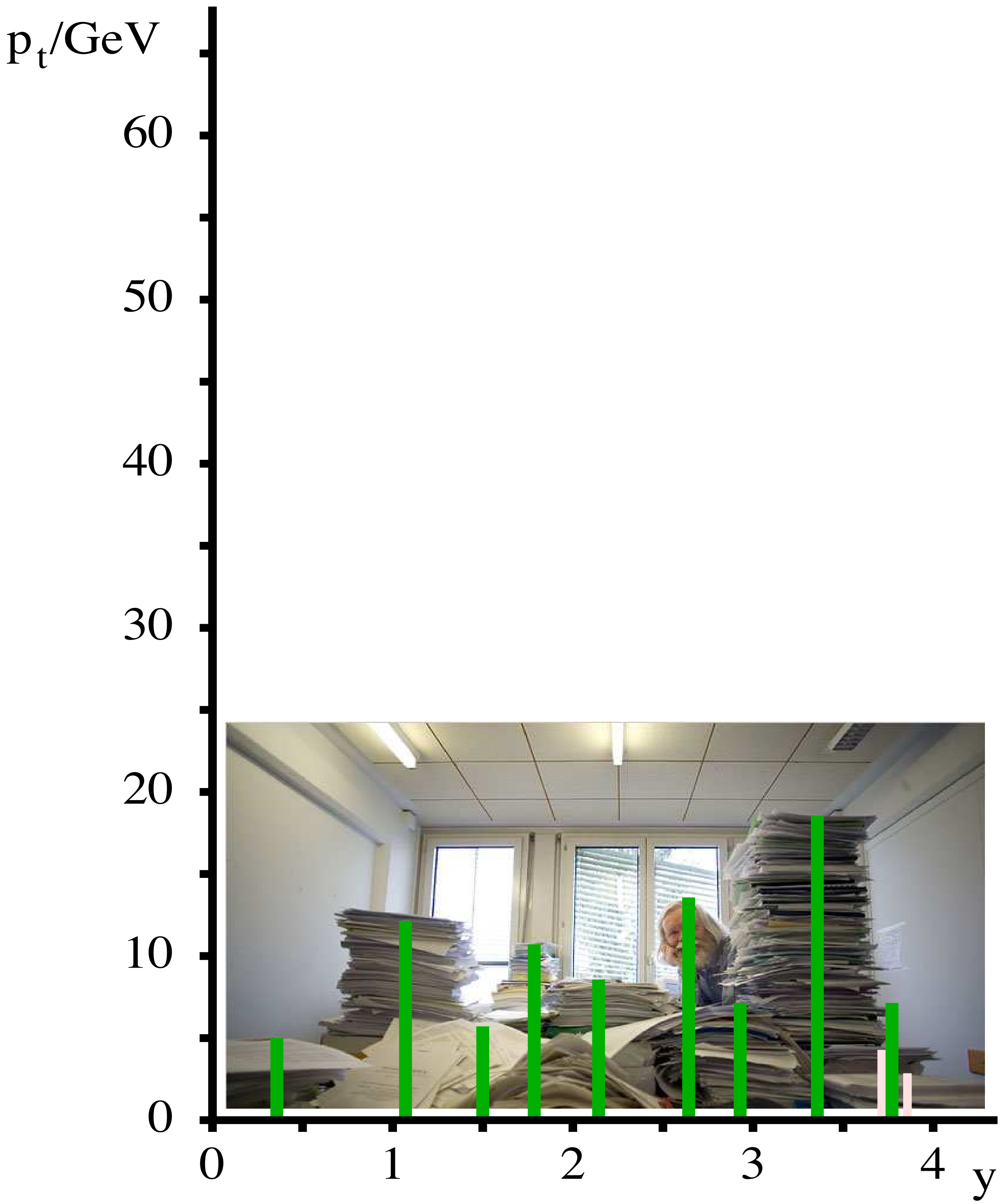
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

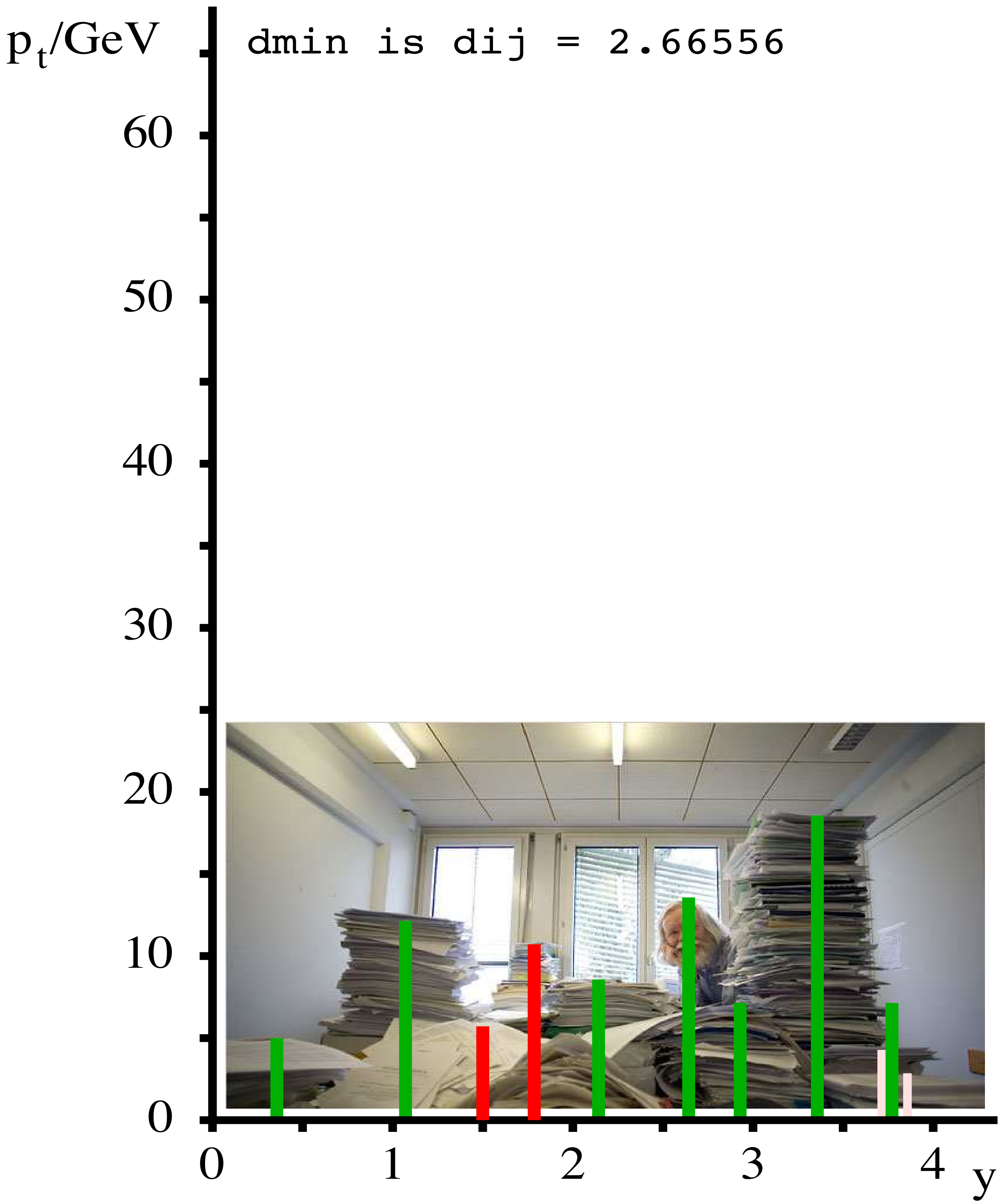
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

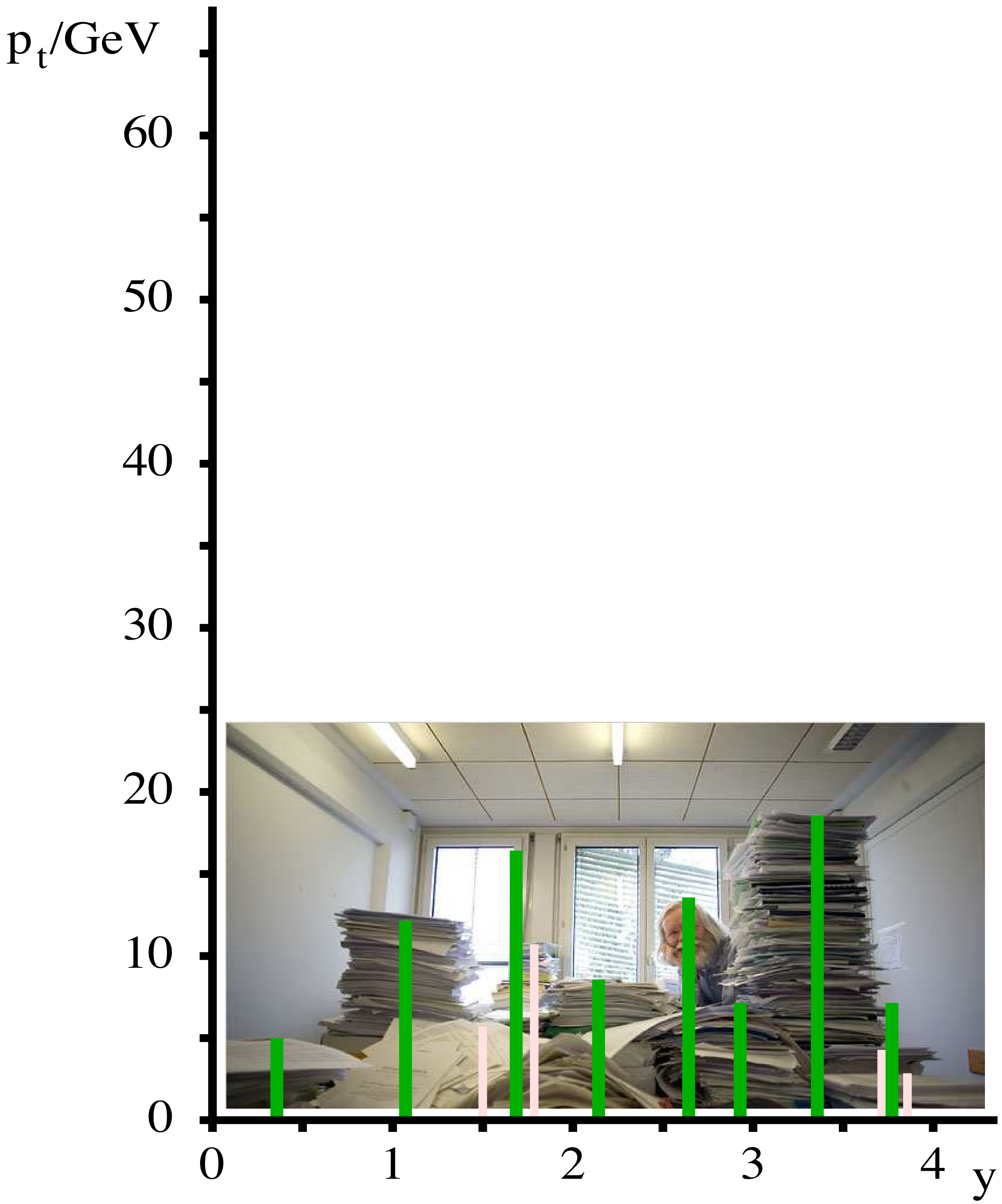
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

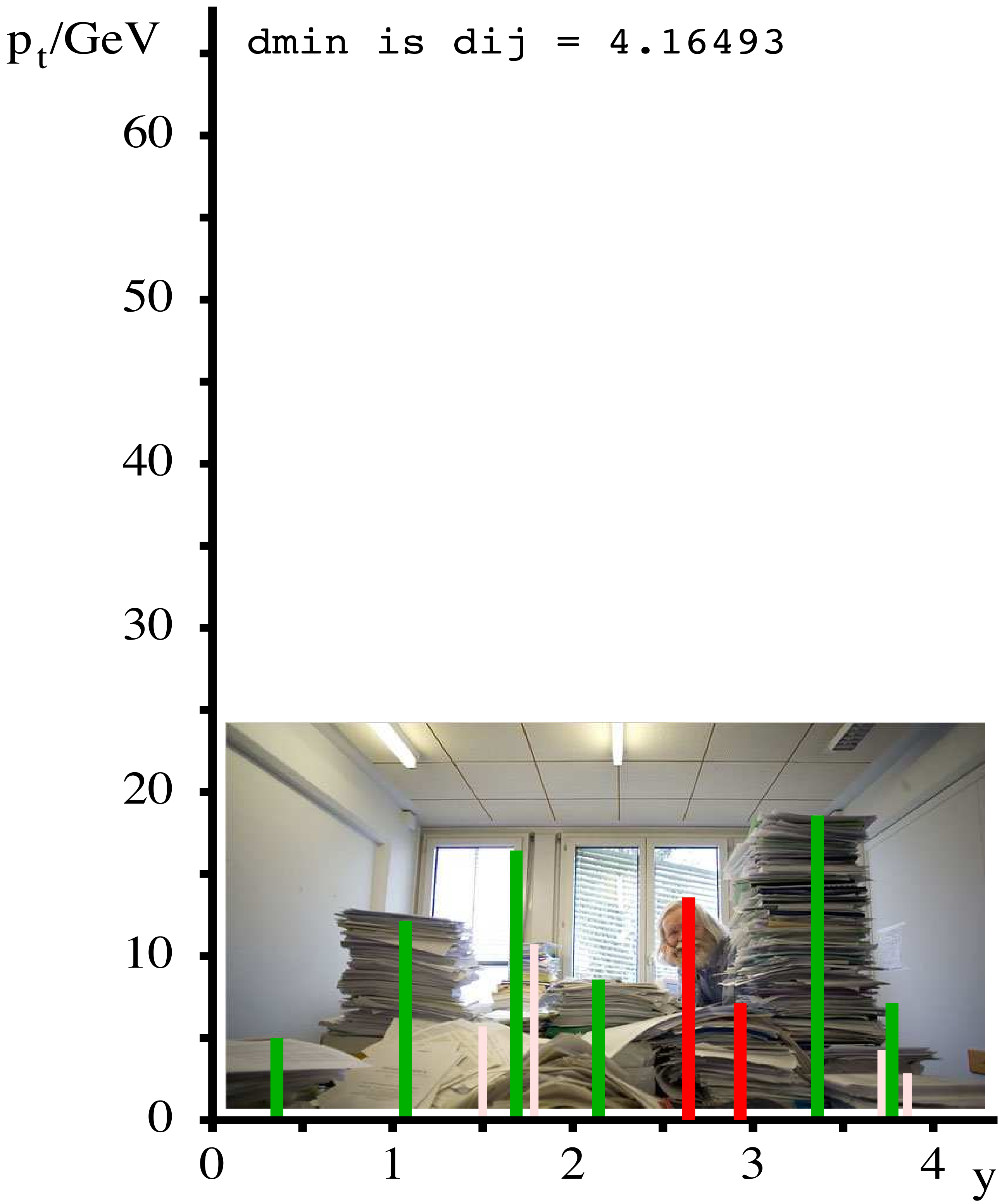
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

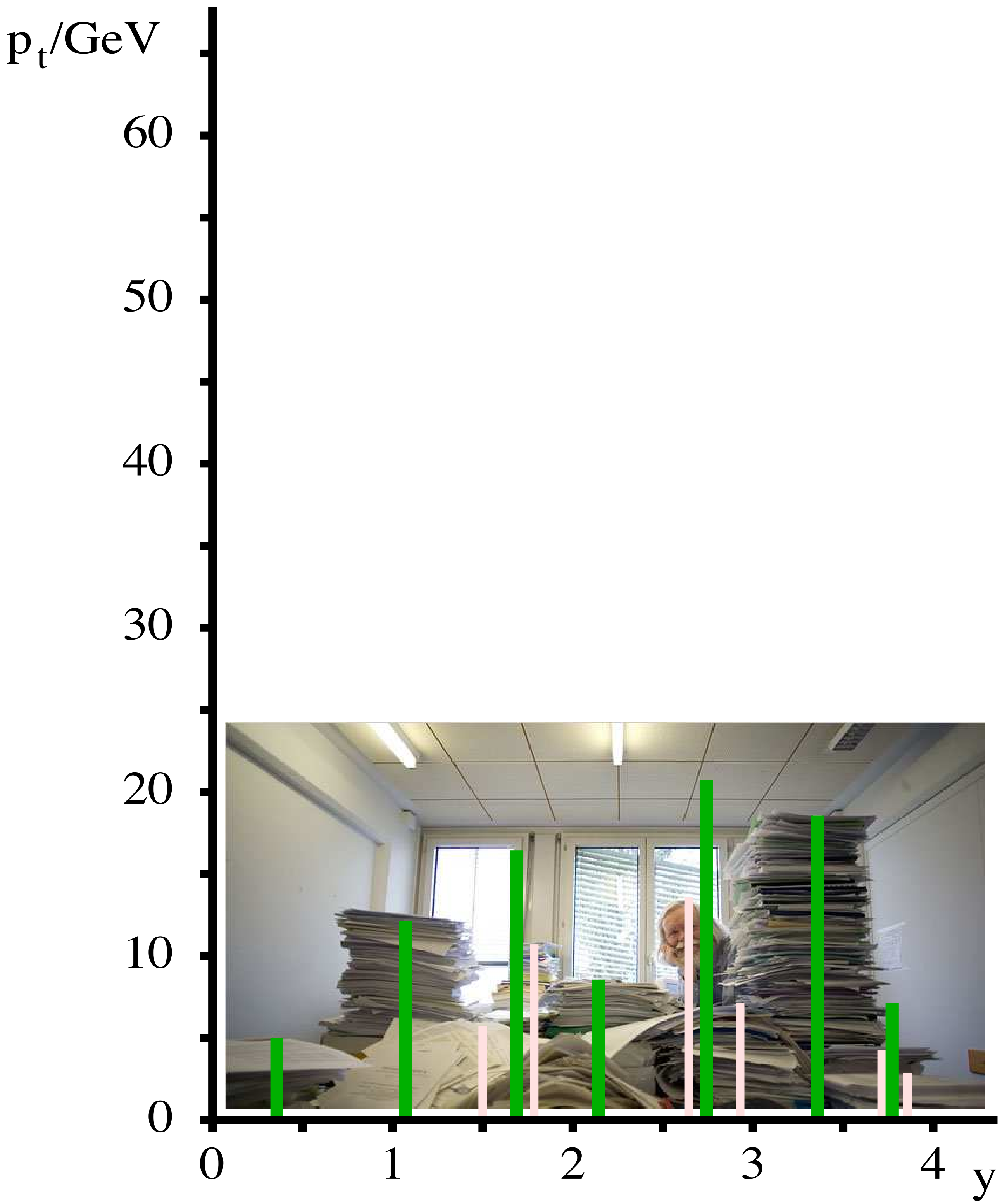
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

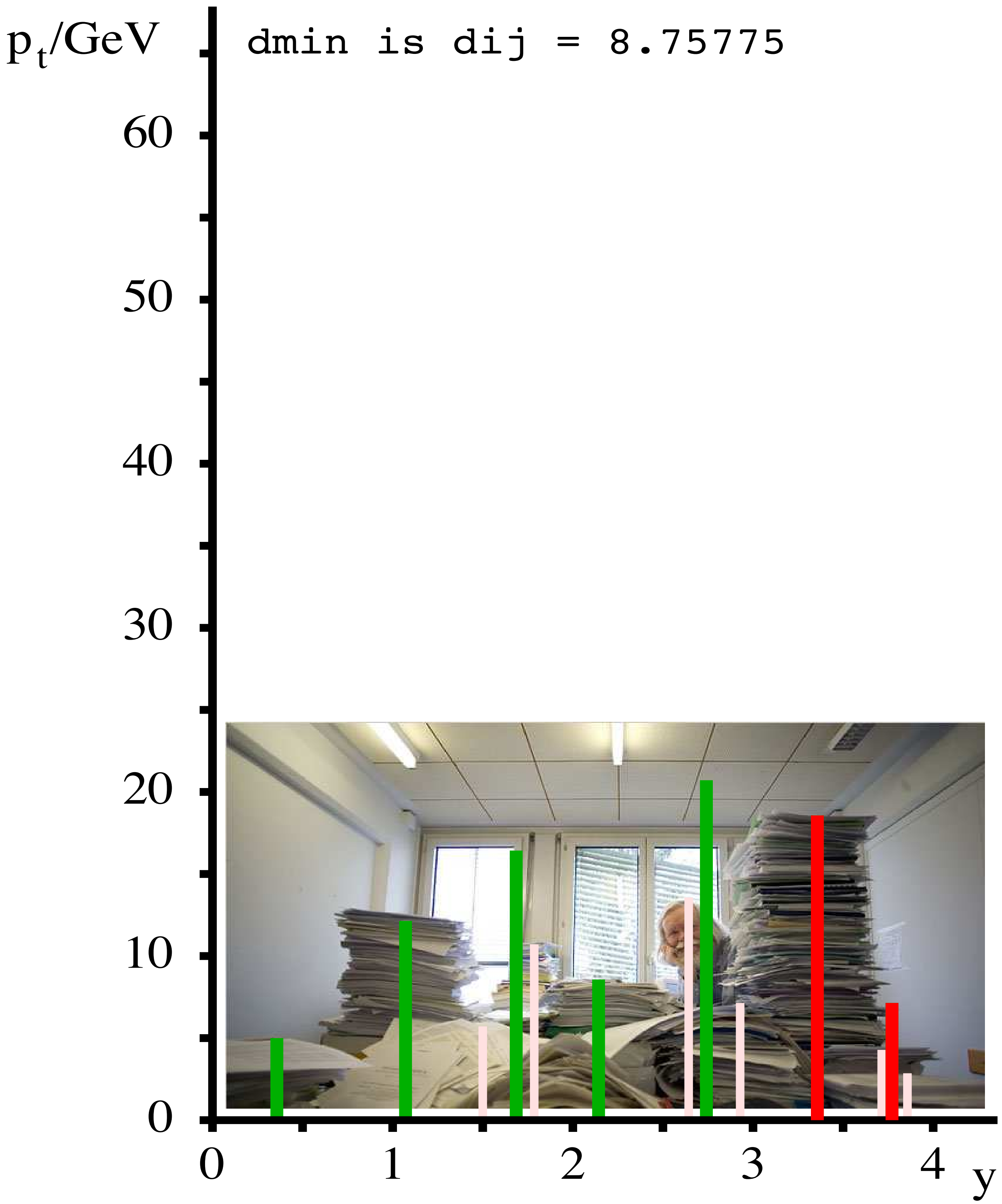
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

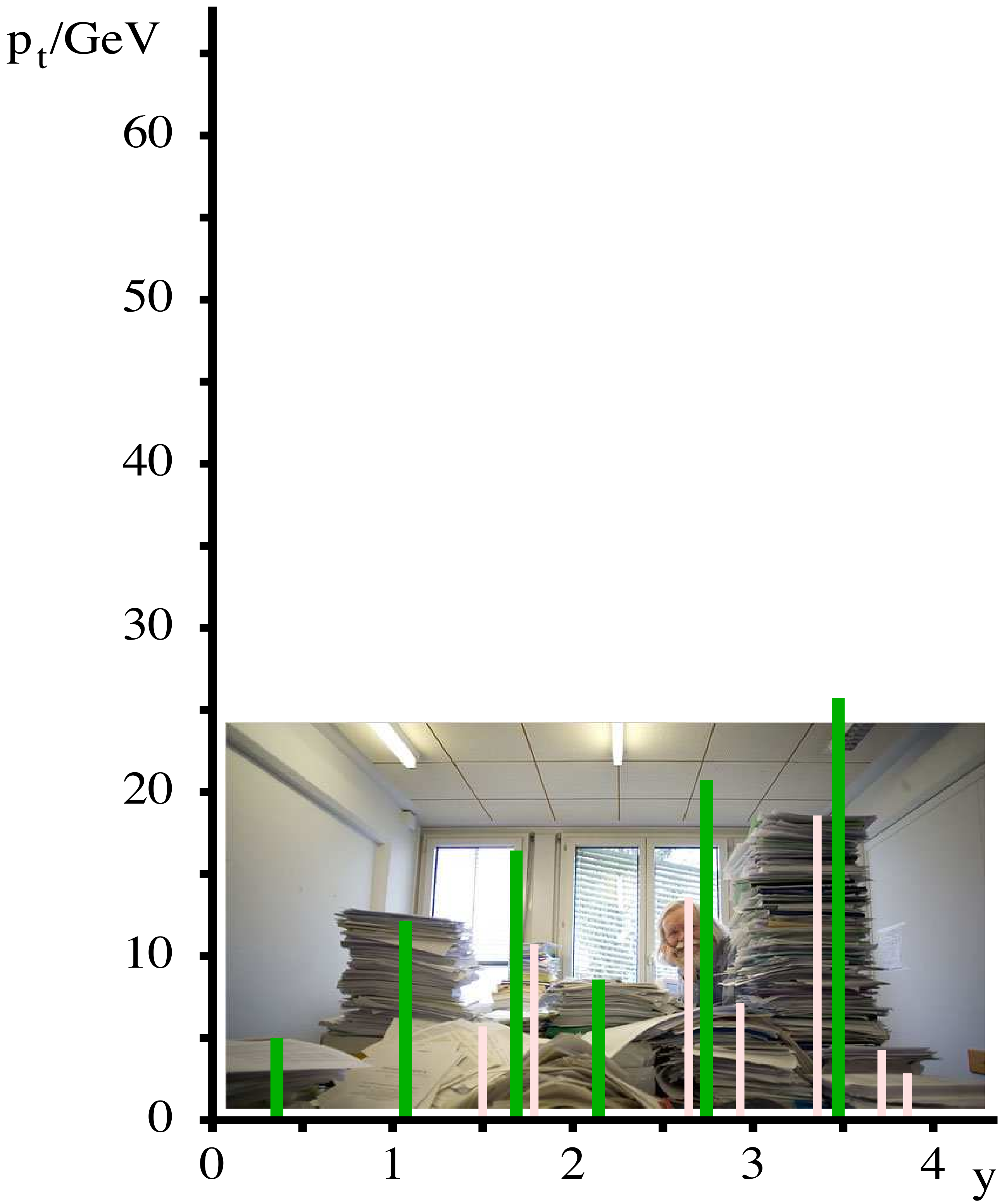
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

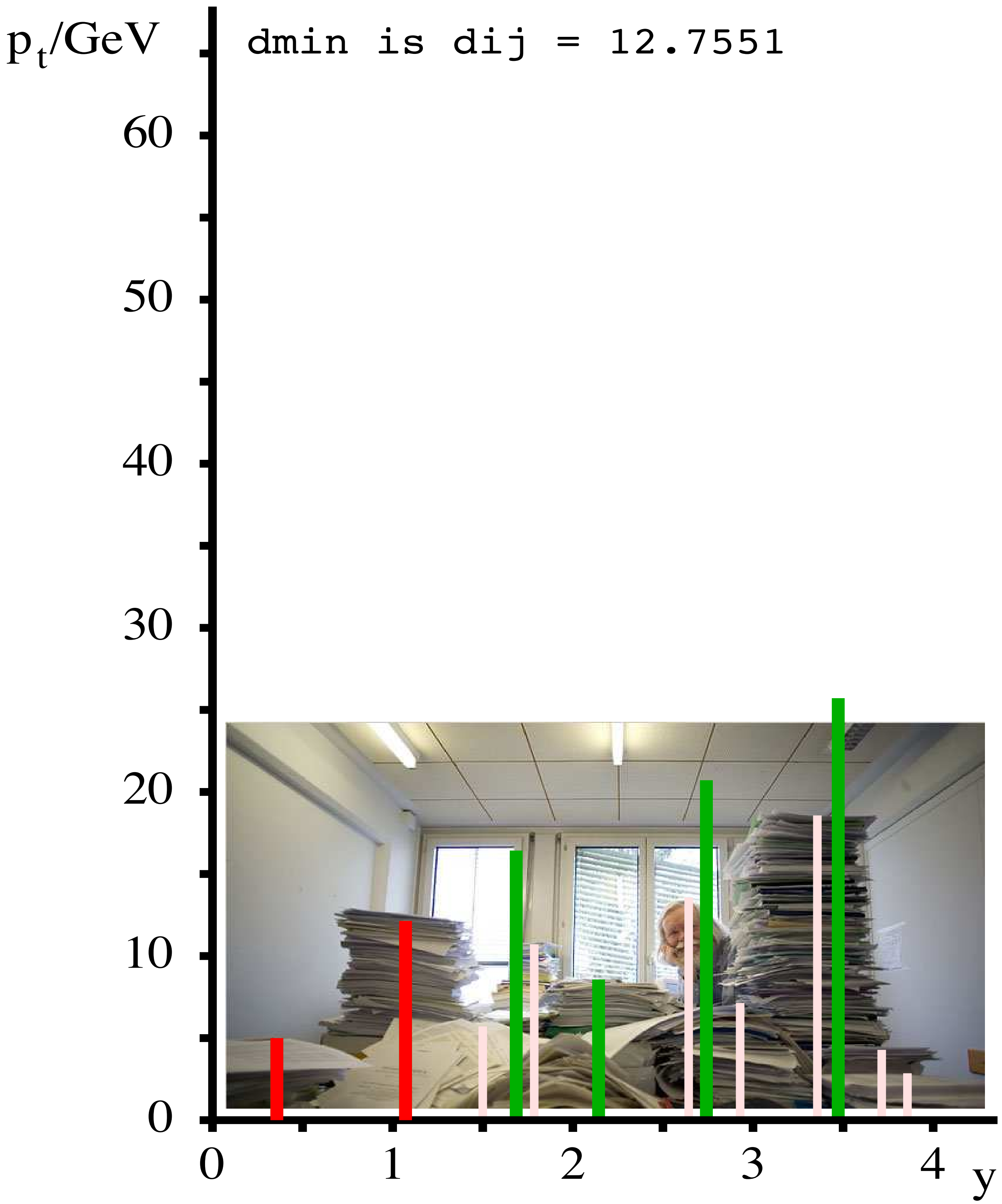
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

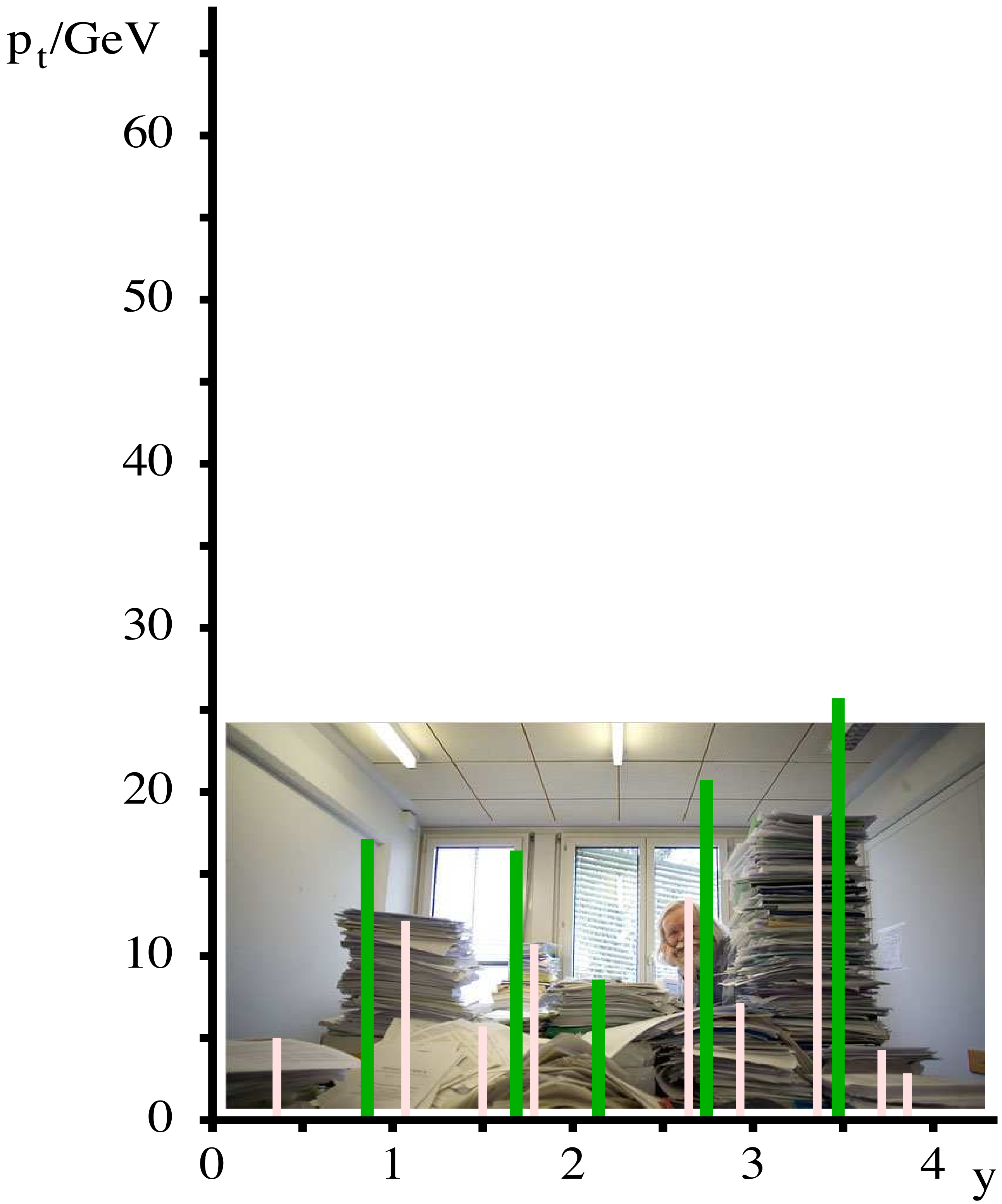
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

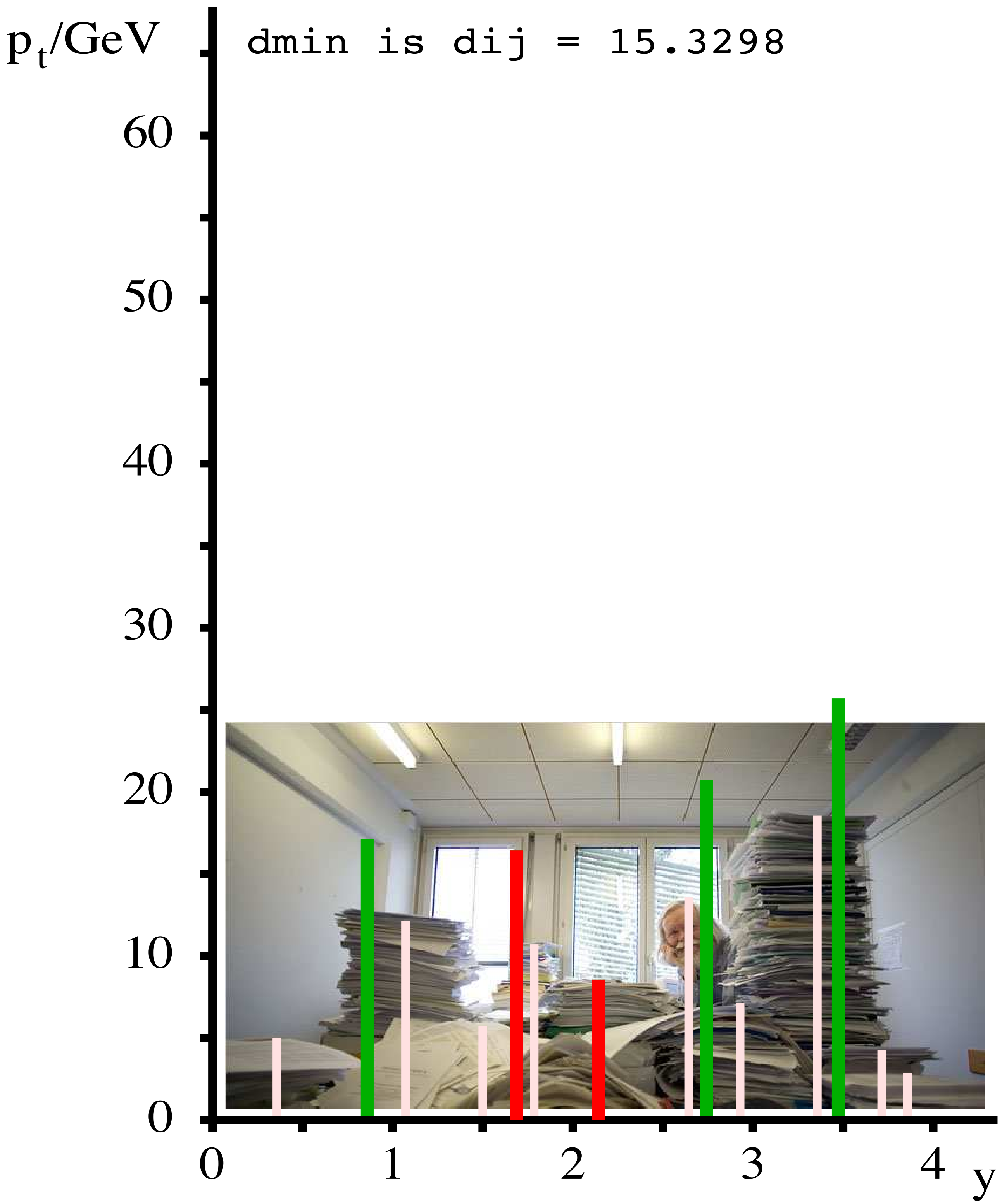
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

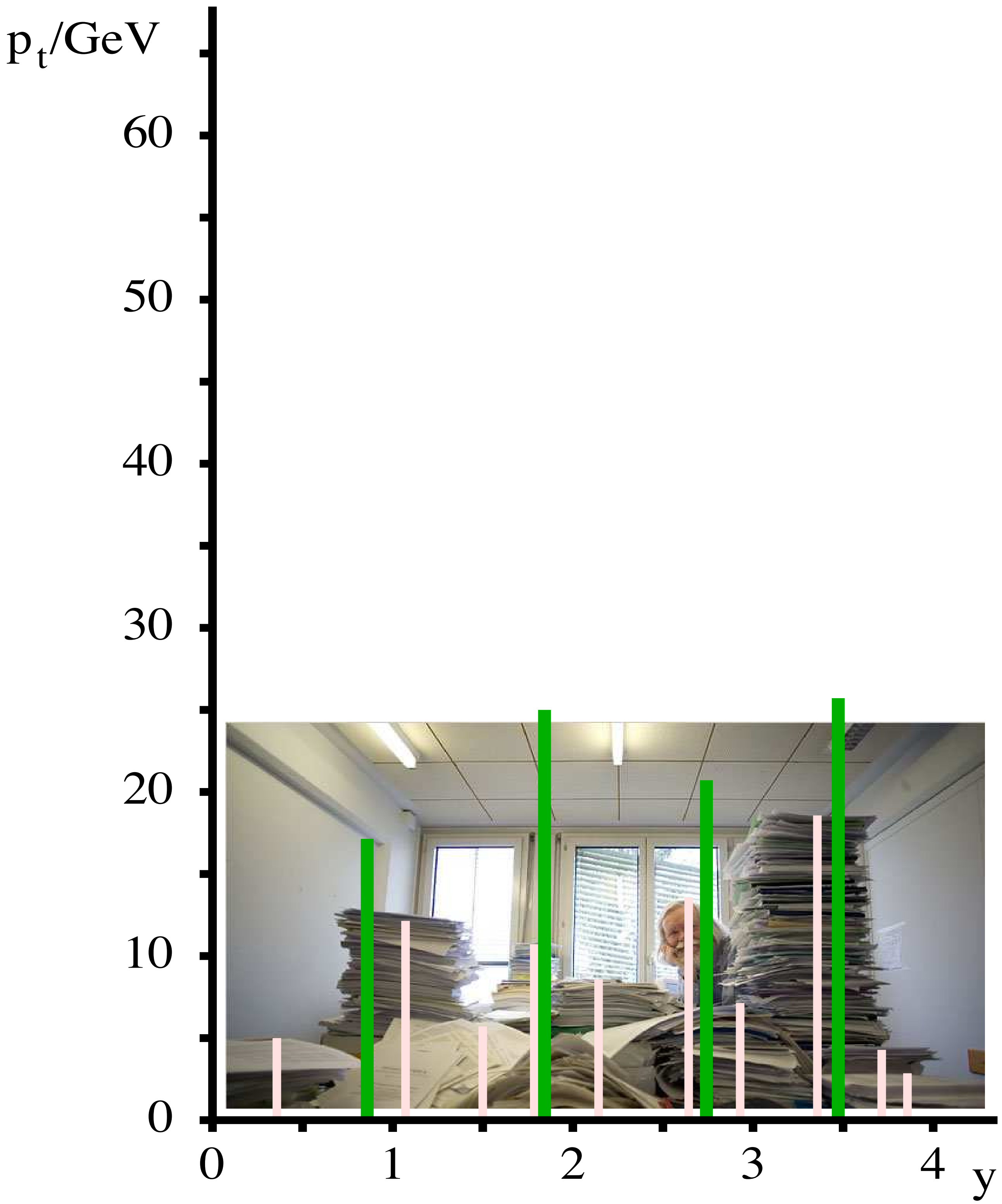
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

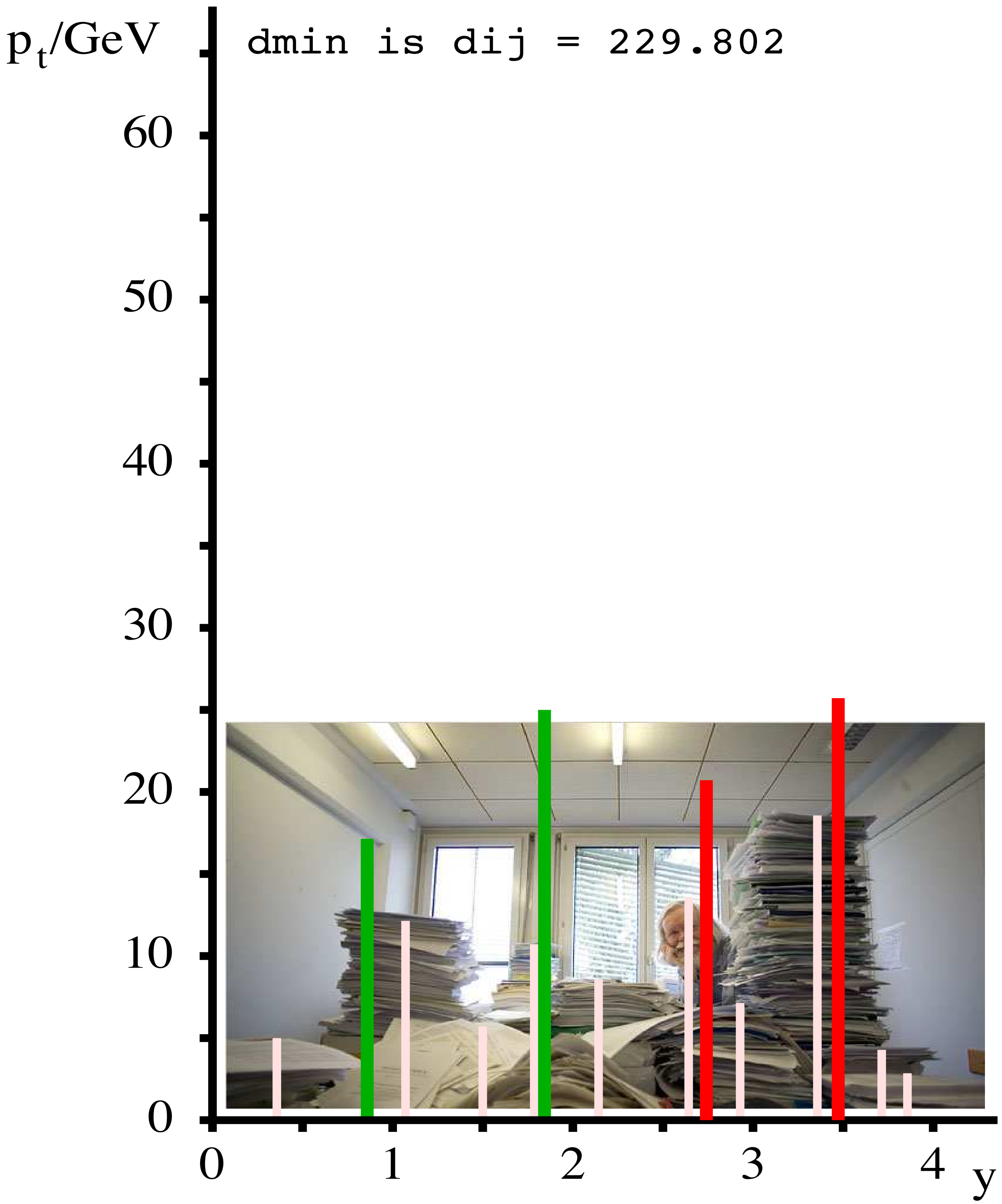
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

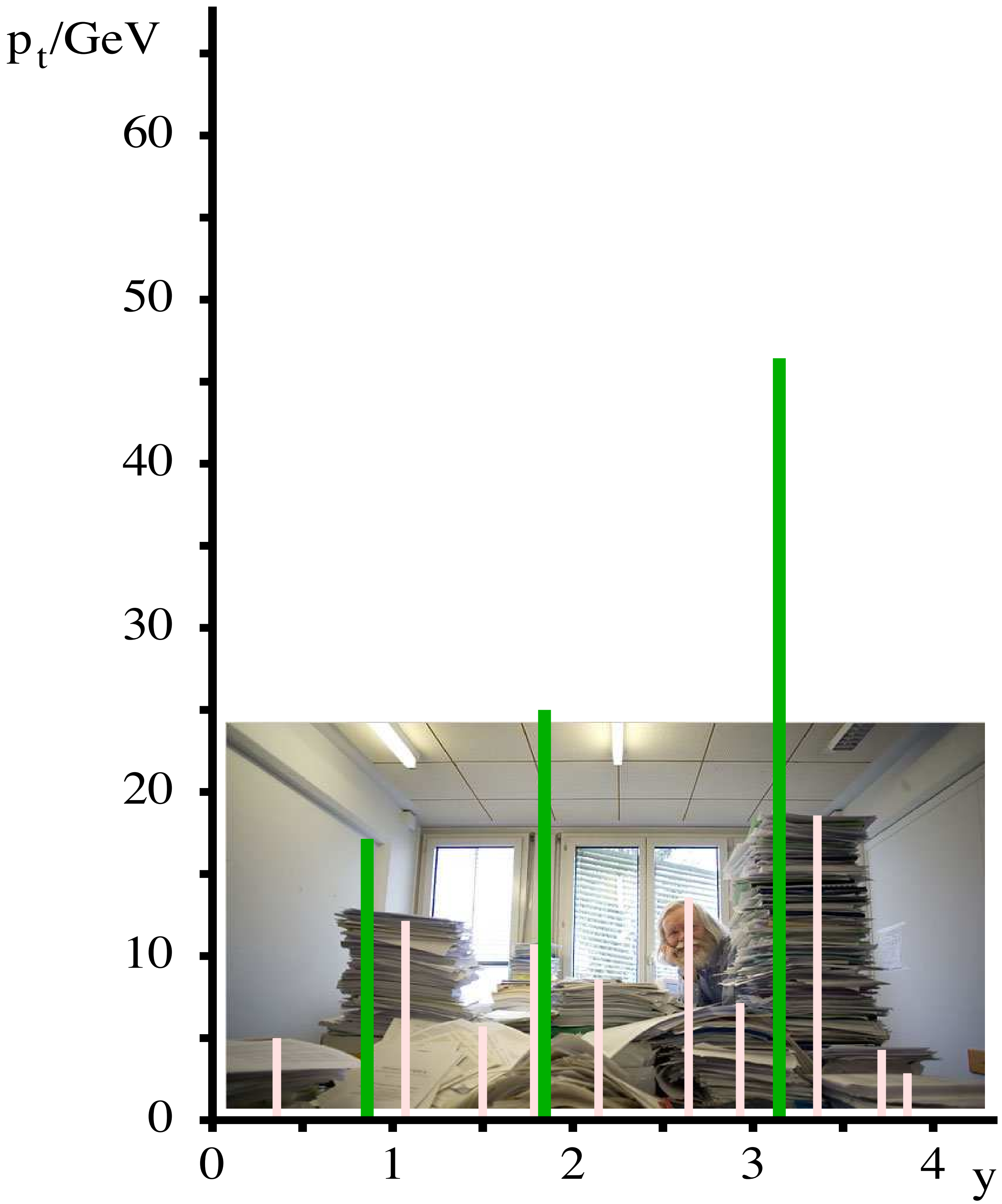
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

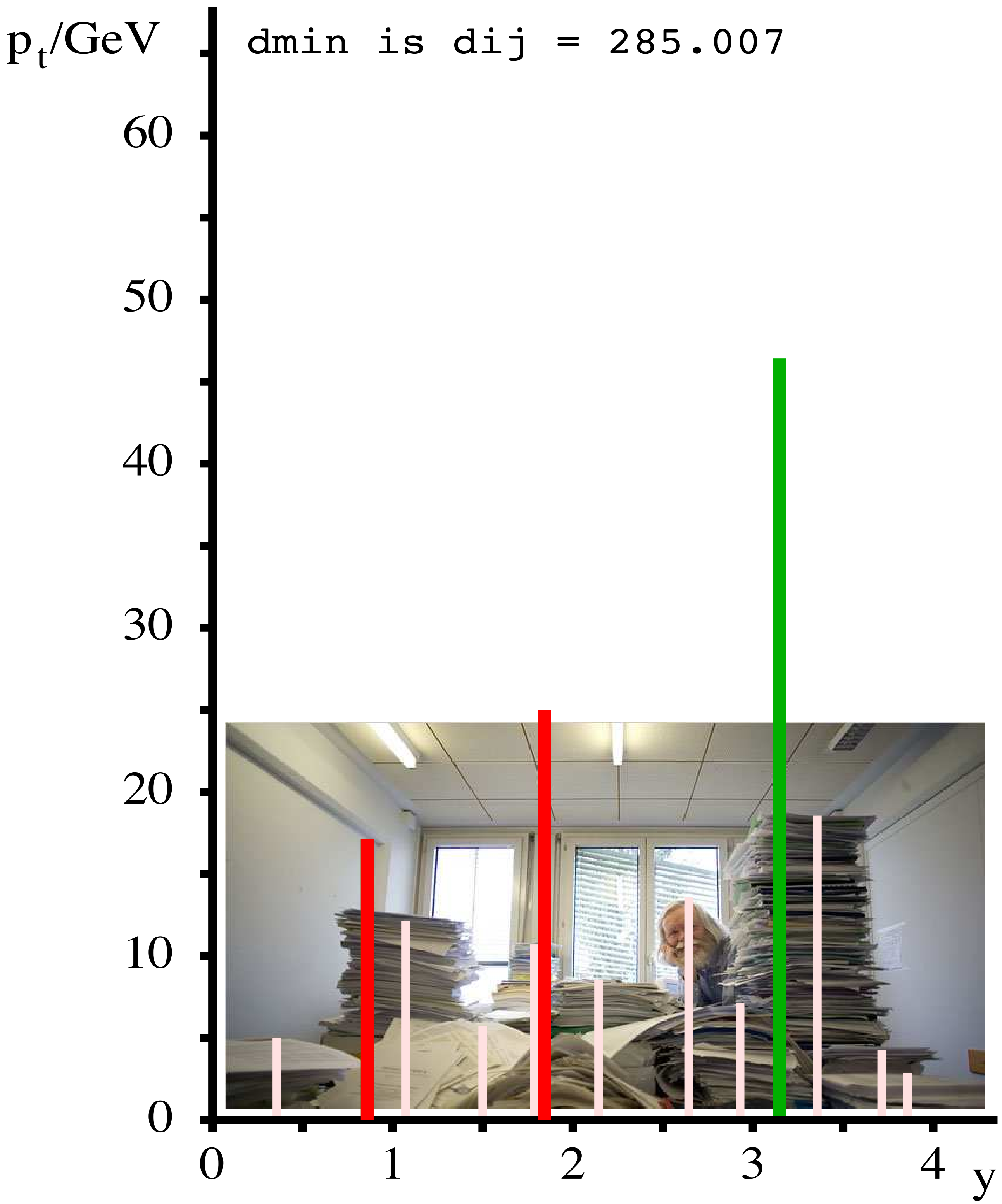
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

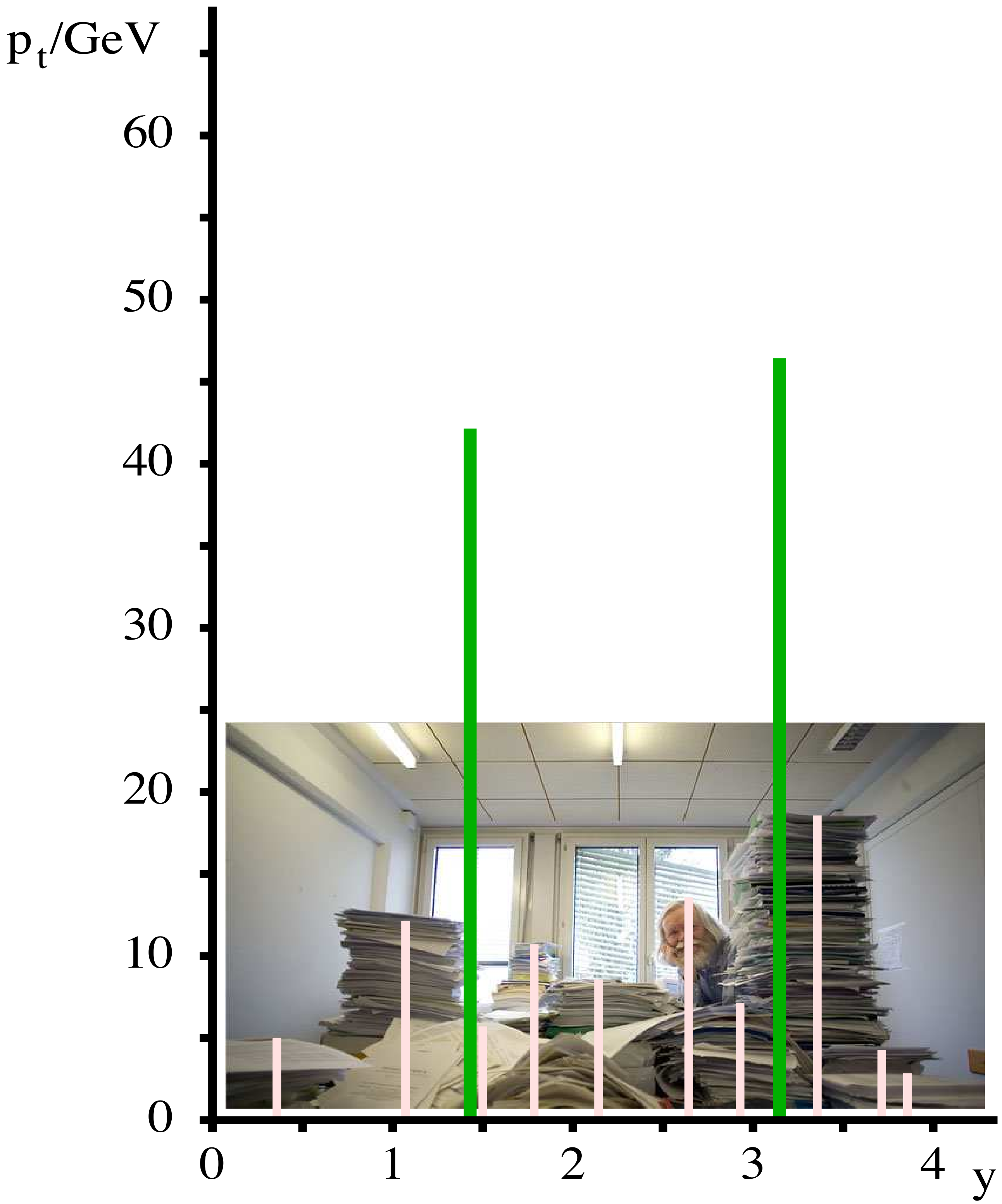
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

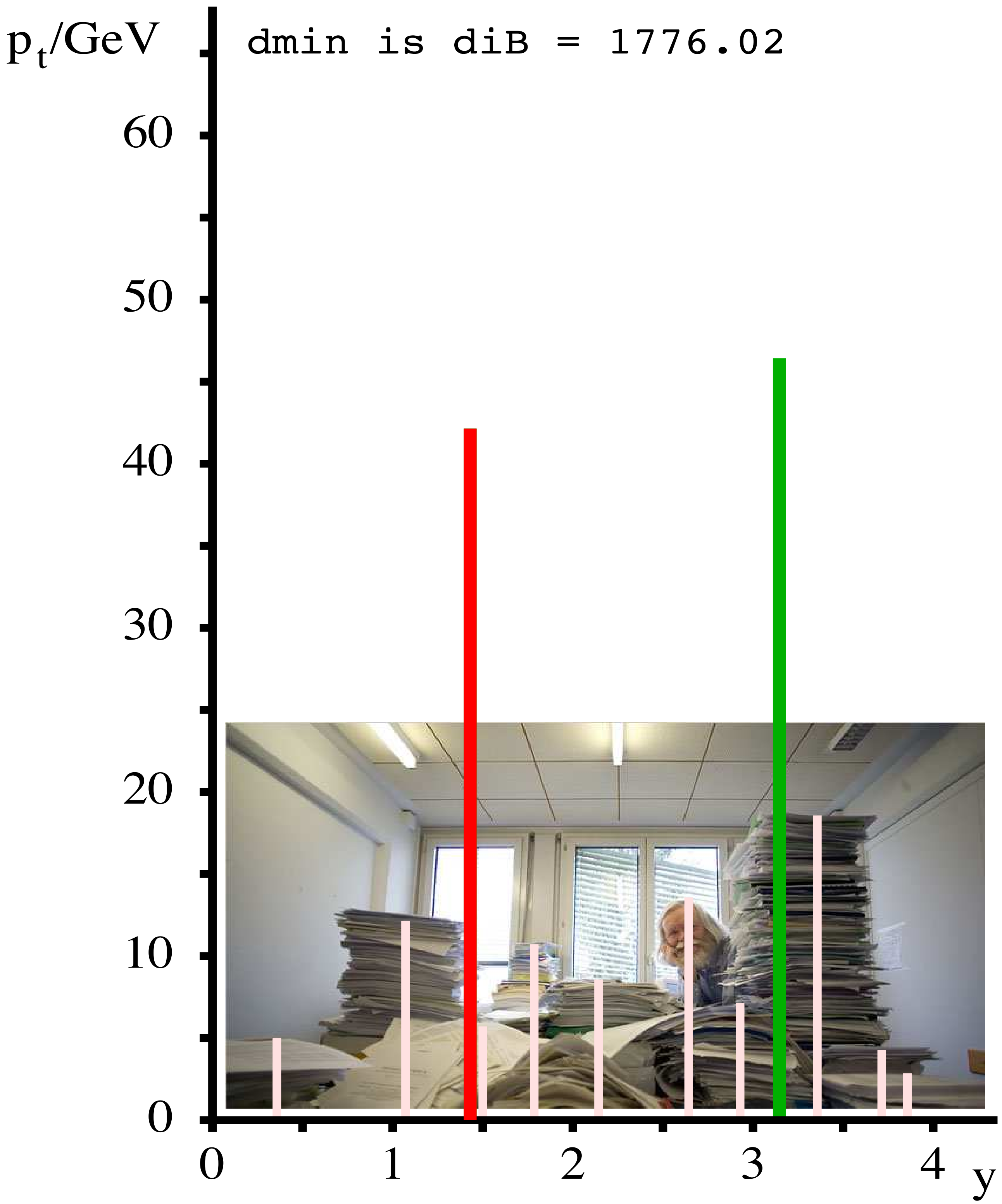
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

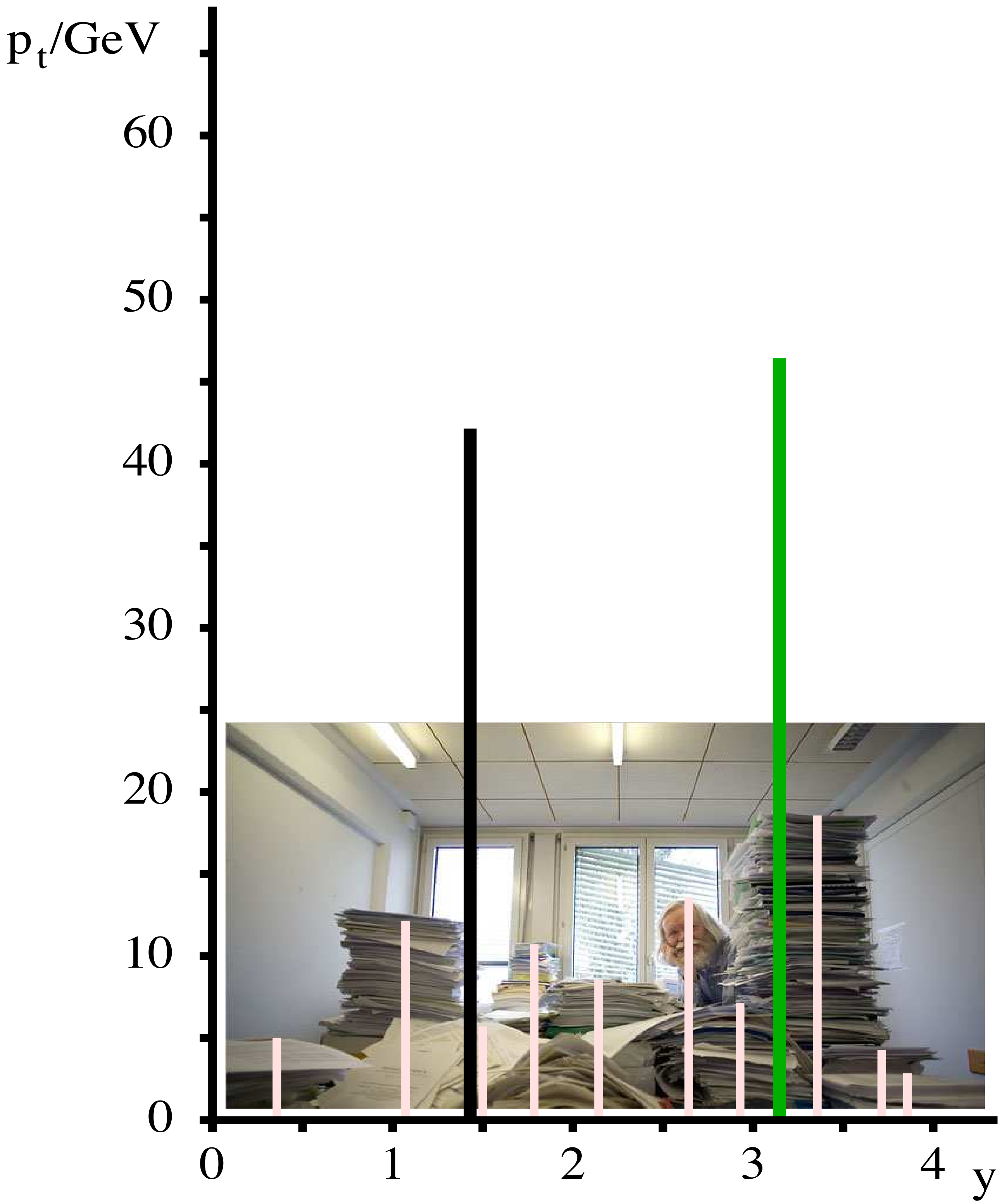
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

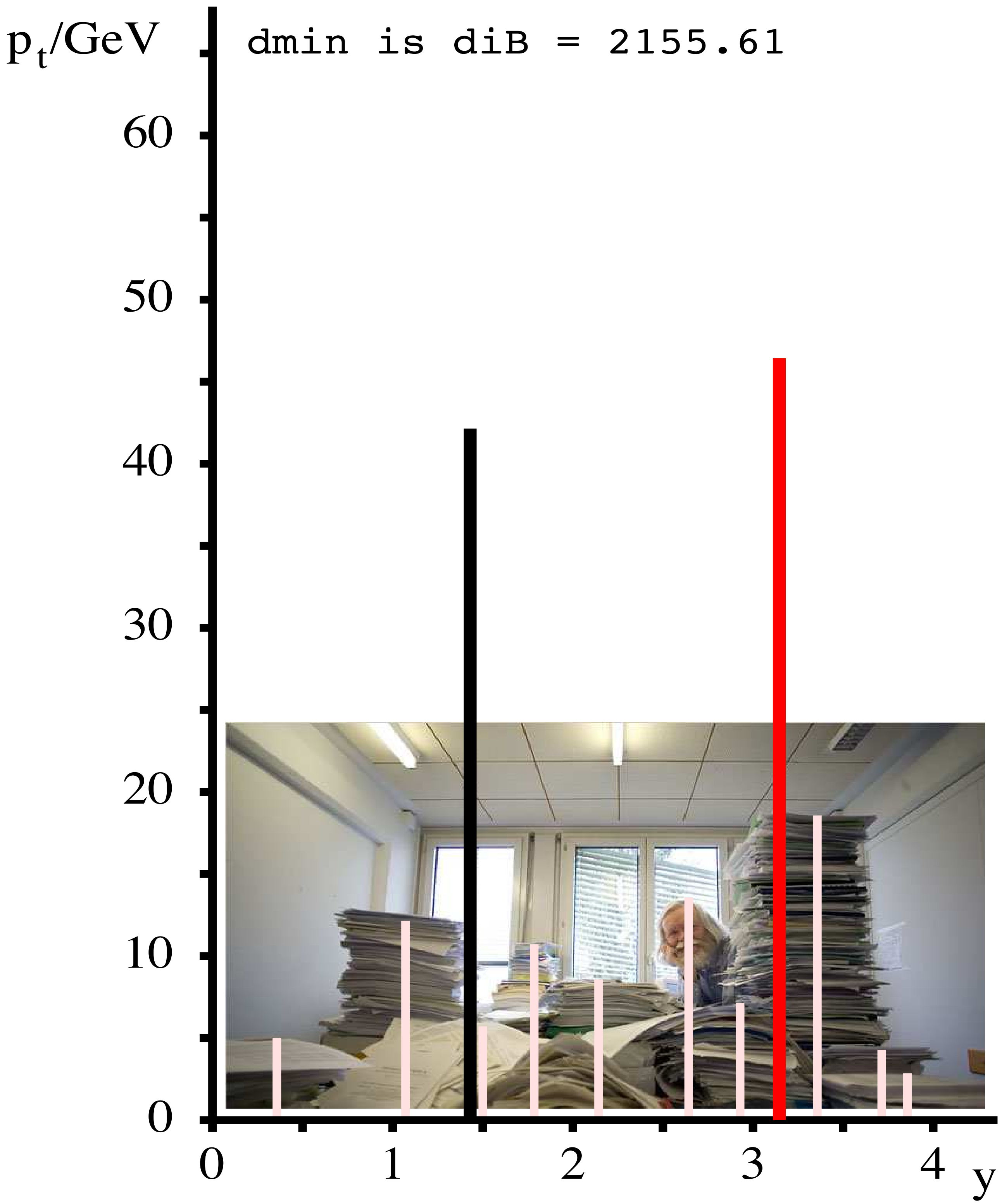
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

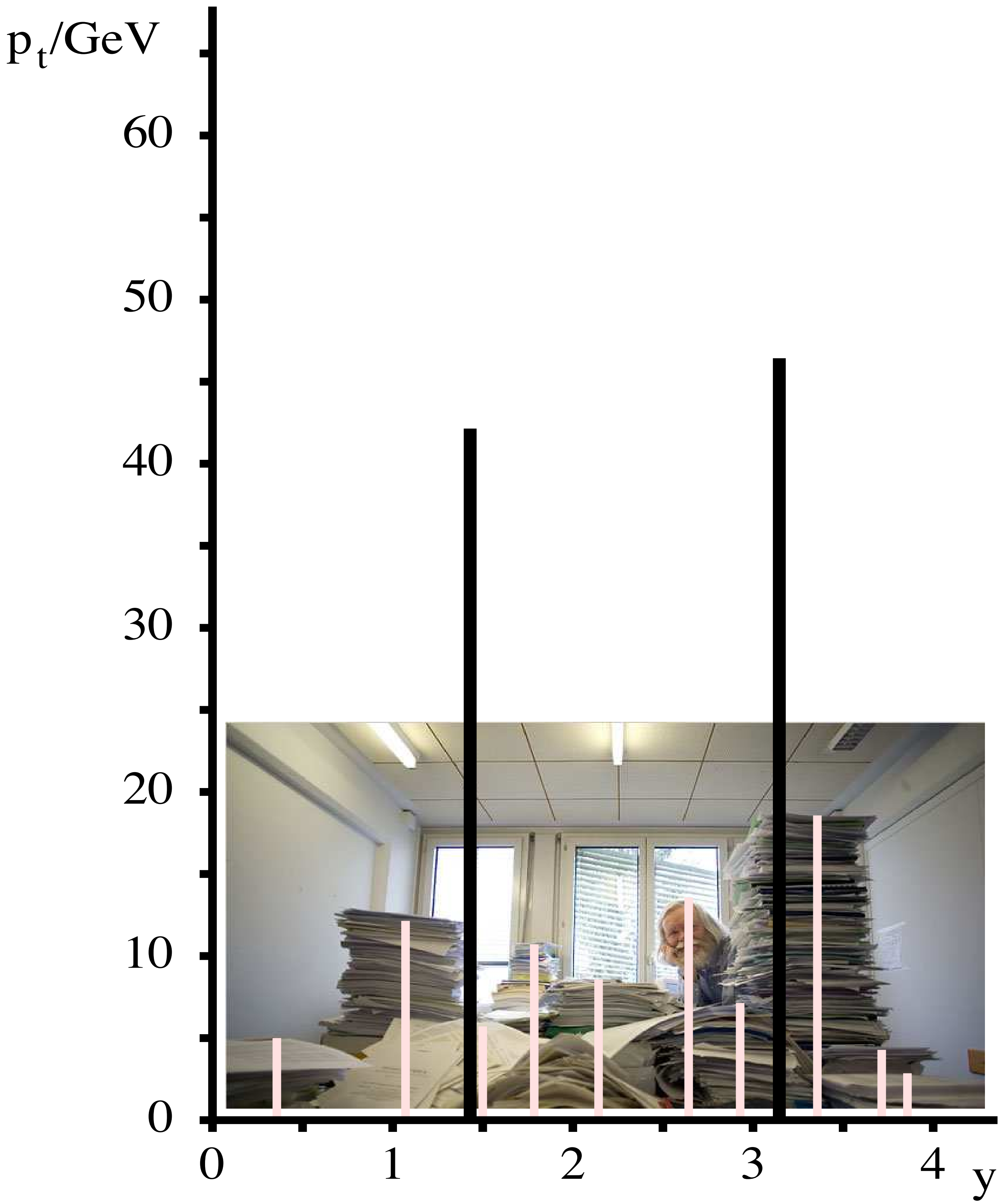
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

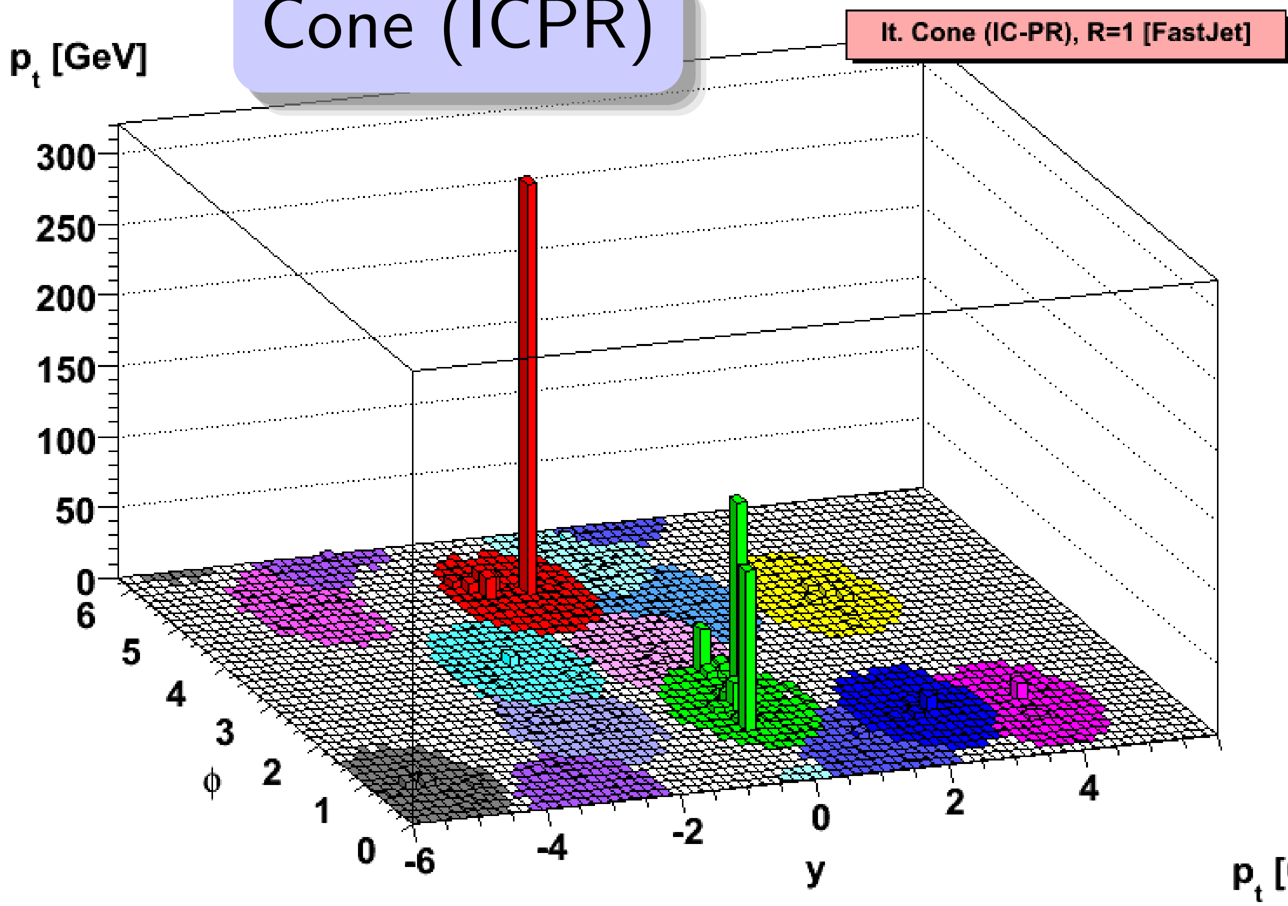
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

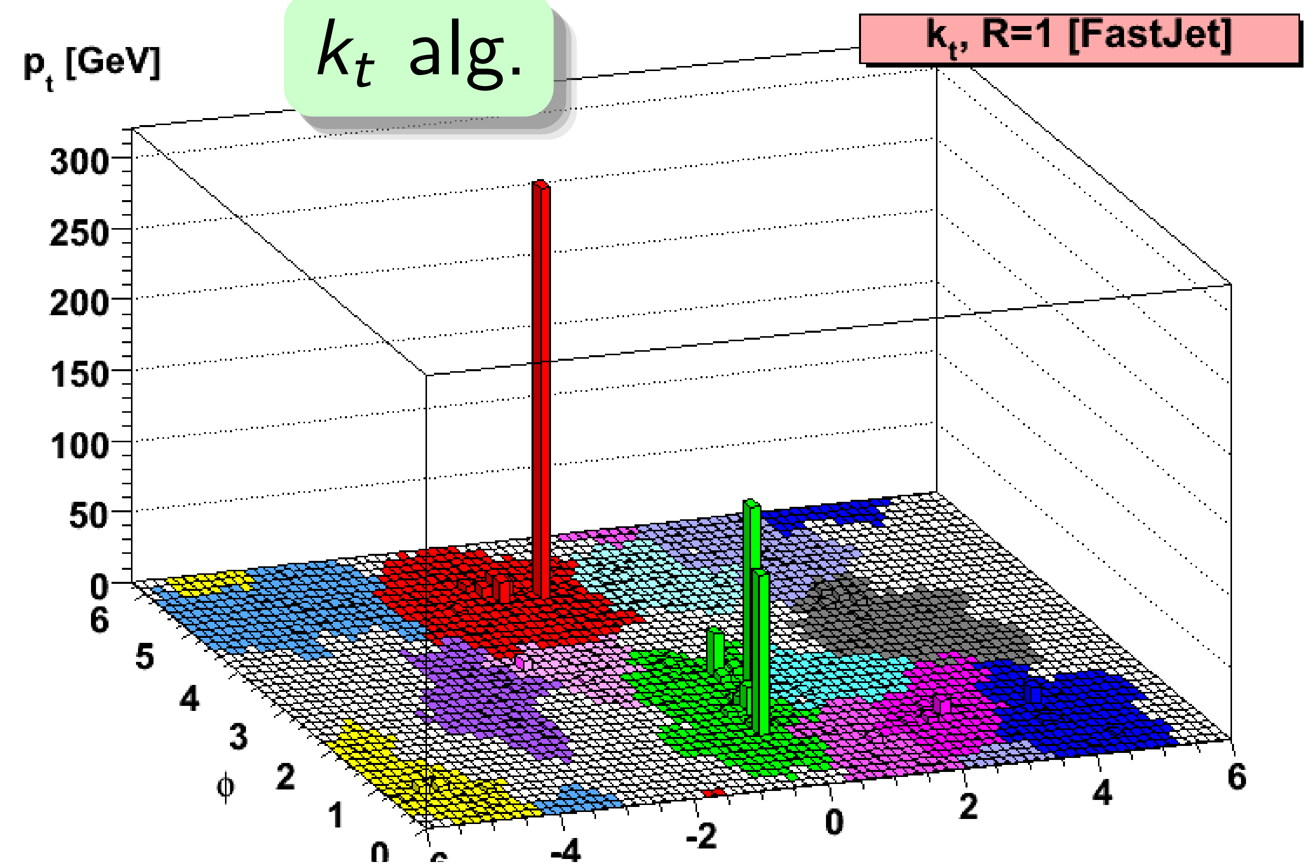
Cone (ICPR)



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

Much appreciated by experiments e.g. for acceptance corrections

k_t alg.



k_t jets are **irregular**

Because soft junk clusters together first:

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

Regularly held against k_t

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

Others differ in

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

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

Sequential recombination variants

Cambridge/Aachen: the simplest of hadron-collider algorithms

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

Dokshitzer, Leder, Moretti, Webber '97 (Cambridge): more involved e^+e^- form
Wobisch & Wengler '99 (Aachen): simple inclusive hadron-collider form
One still applies a $p_{t,min}$ cut to the jets, as for inclusive k_t

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

Anti-k$_t$: formulated similarly to inclusive k$_t$, but with

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$

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

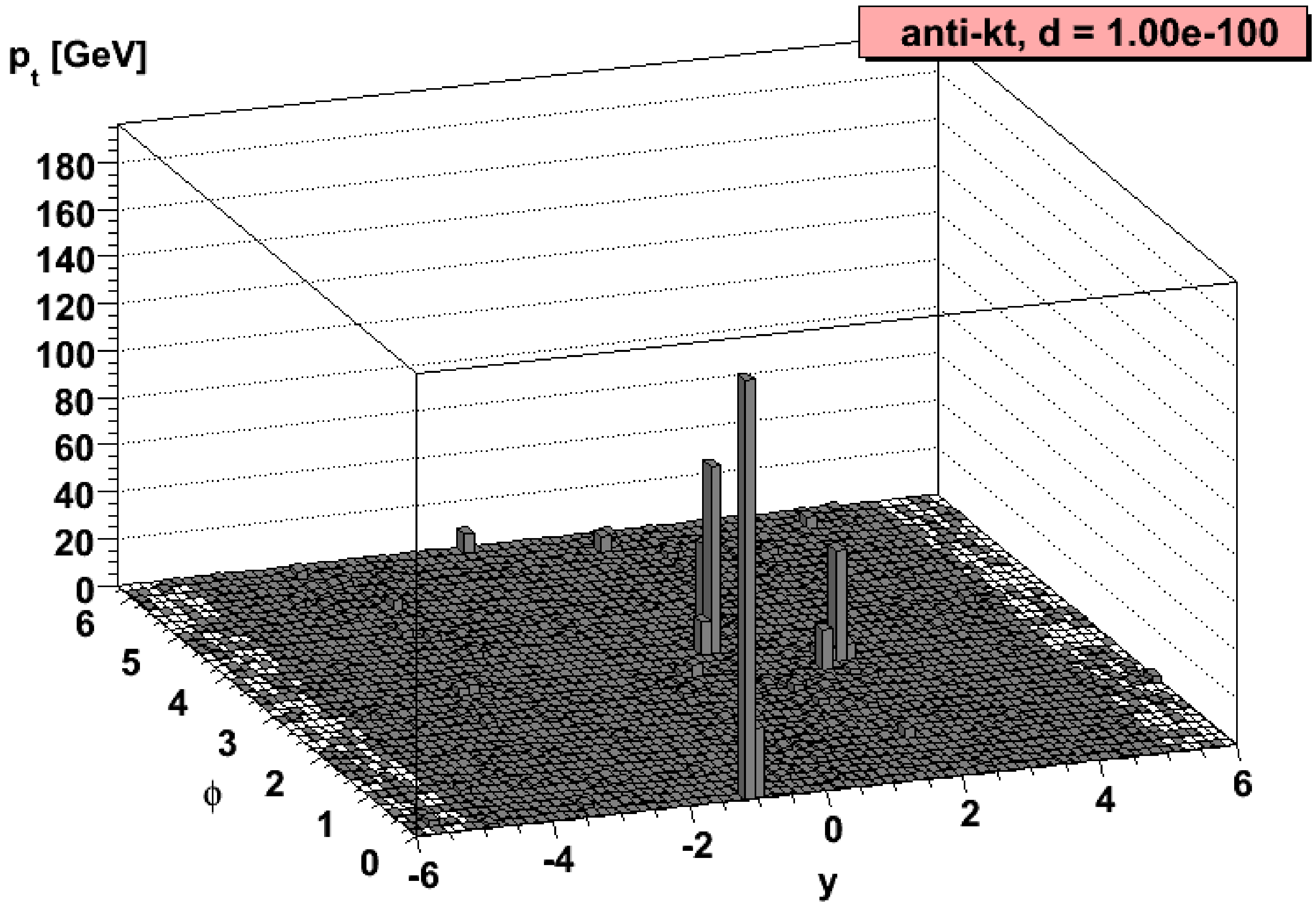
Anti-k$_t$ privileges the collinear divergence of QCD and disfavours clustering between pairs of soft particles

Most pairwise clusterings involve at least one hard particle

Anti-k_t in action

Clustering grows around hard cores

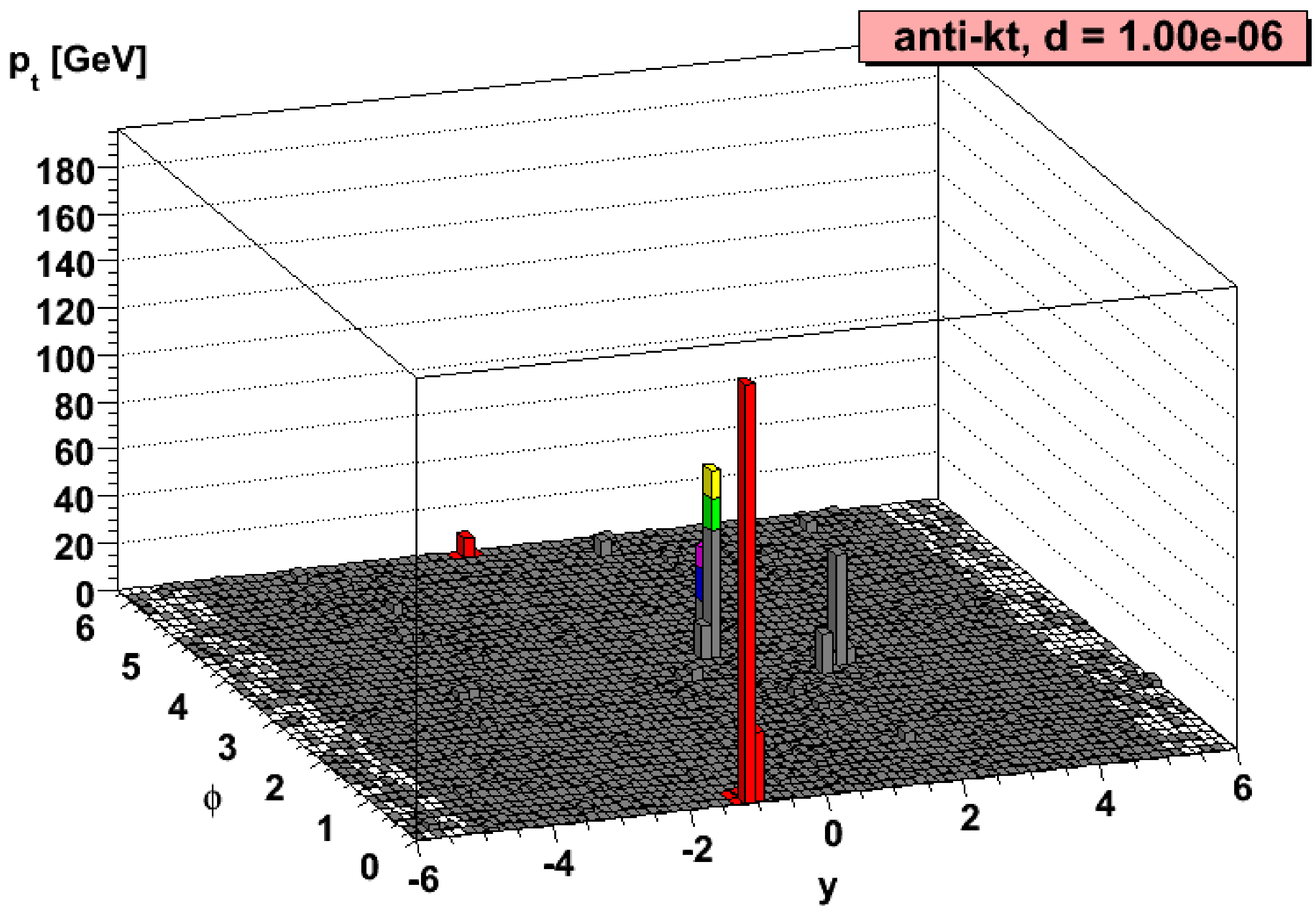
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

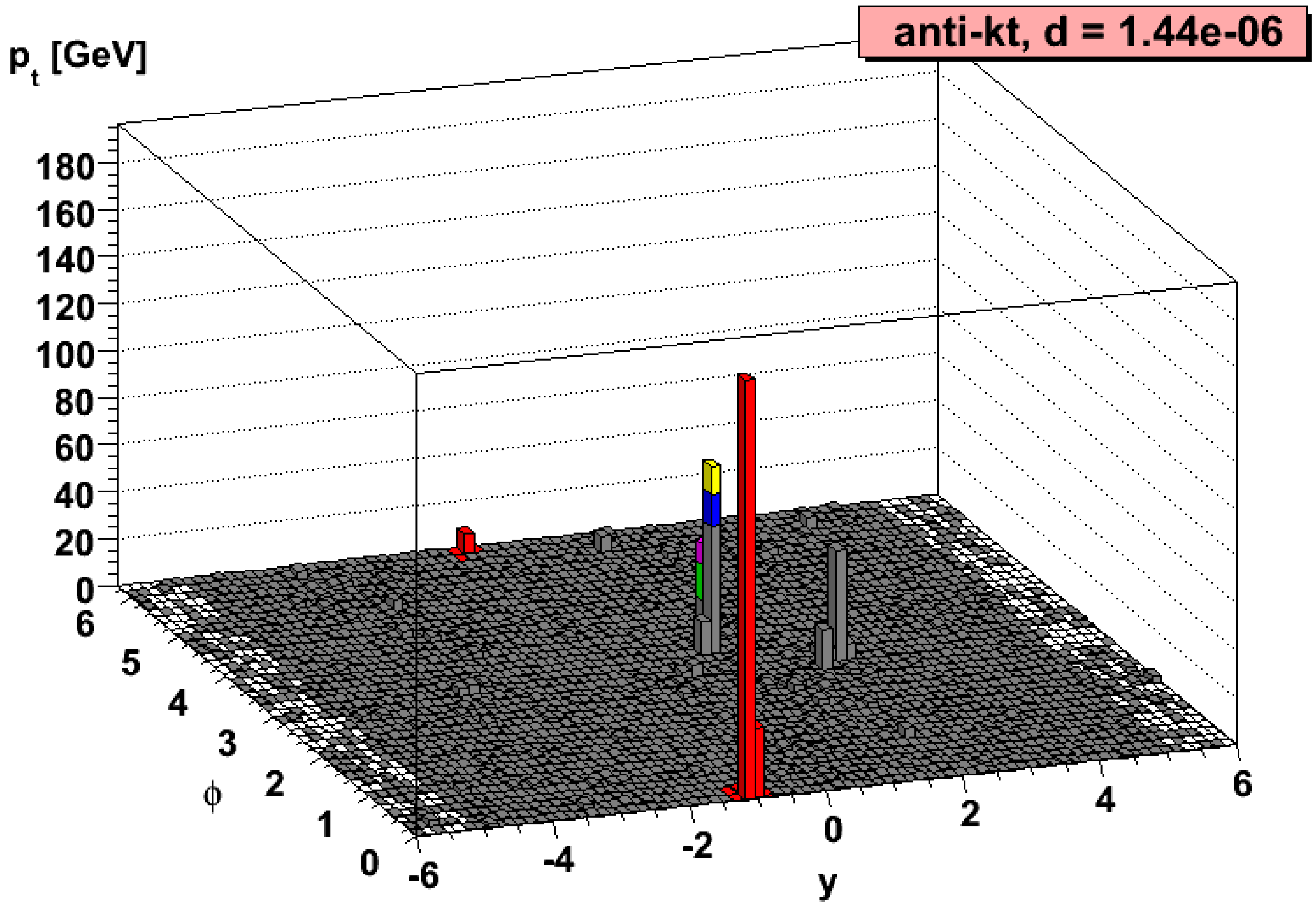
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



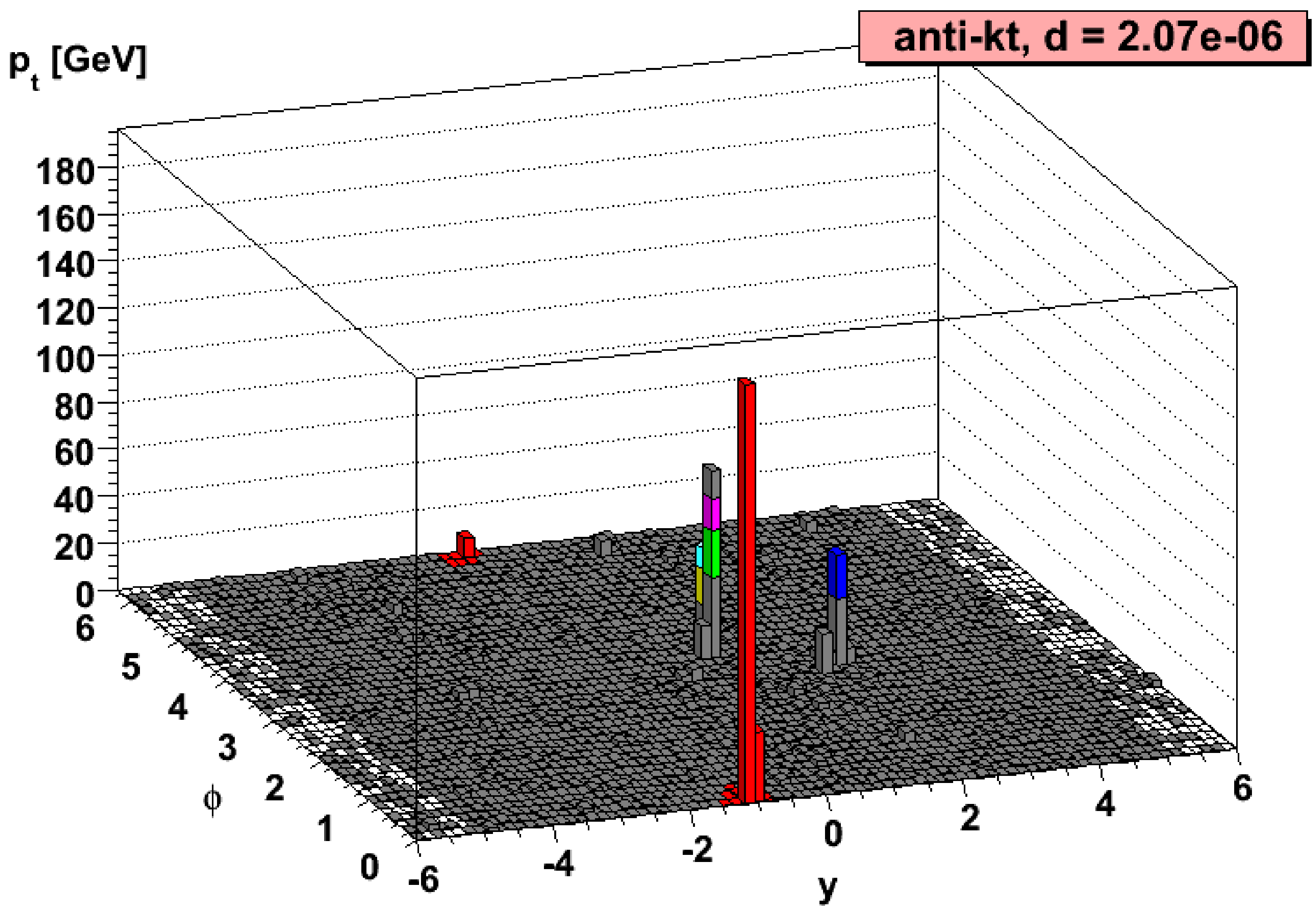
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

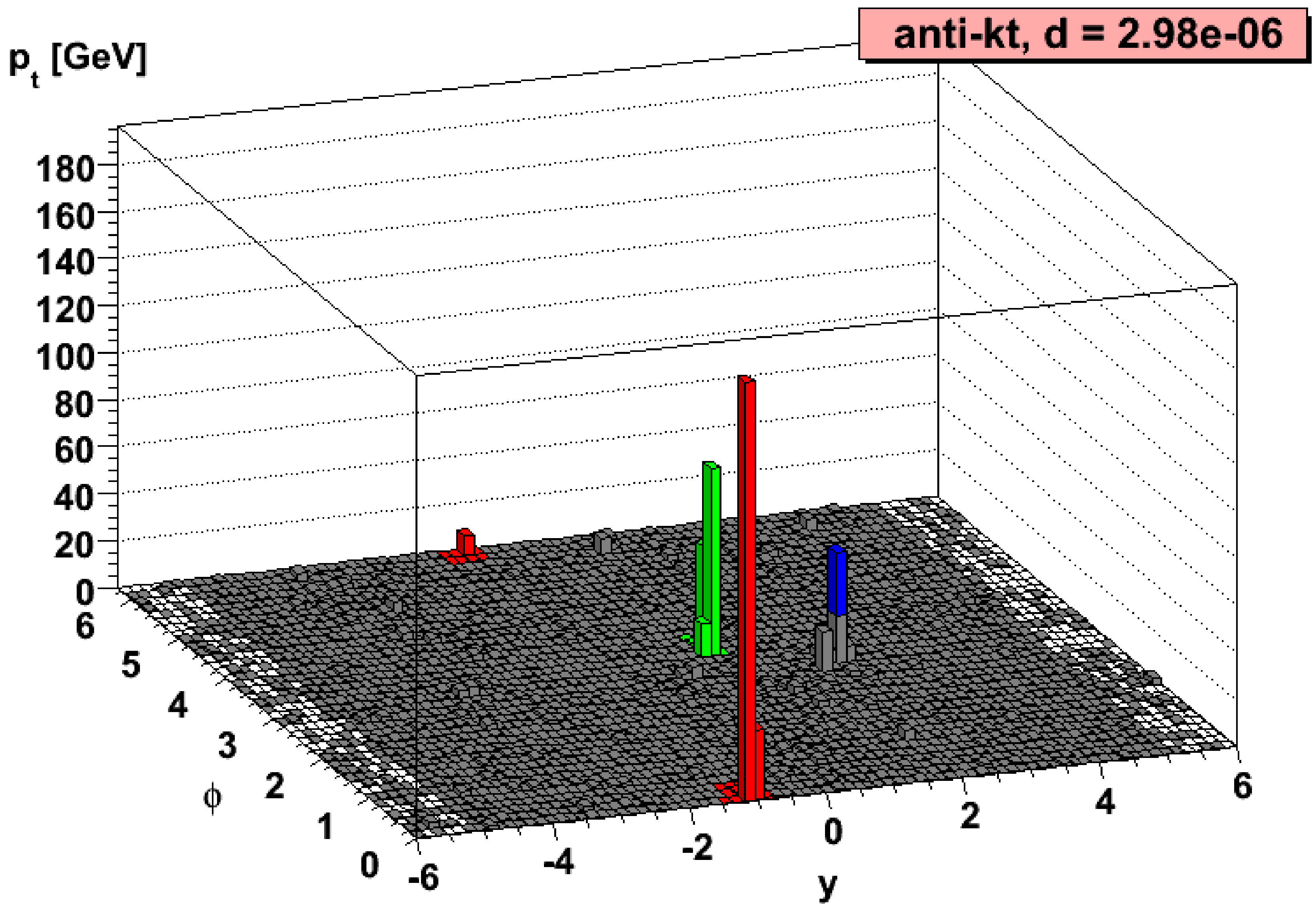
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

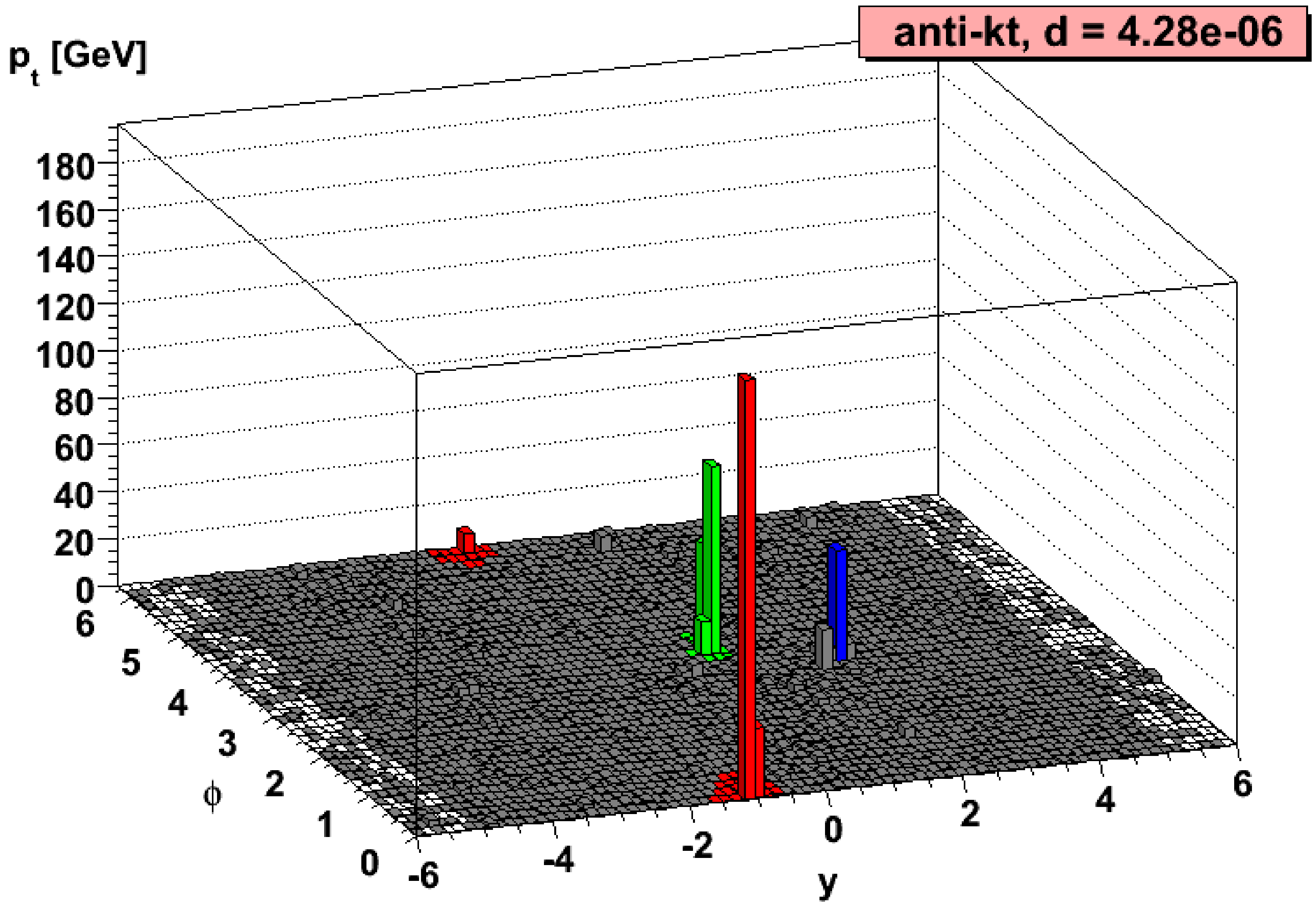
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

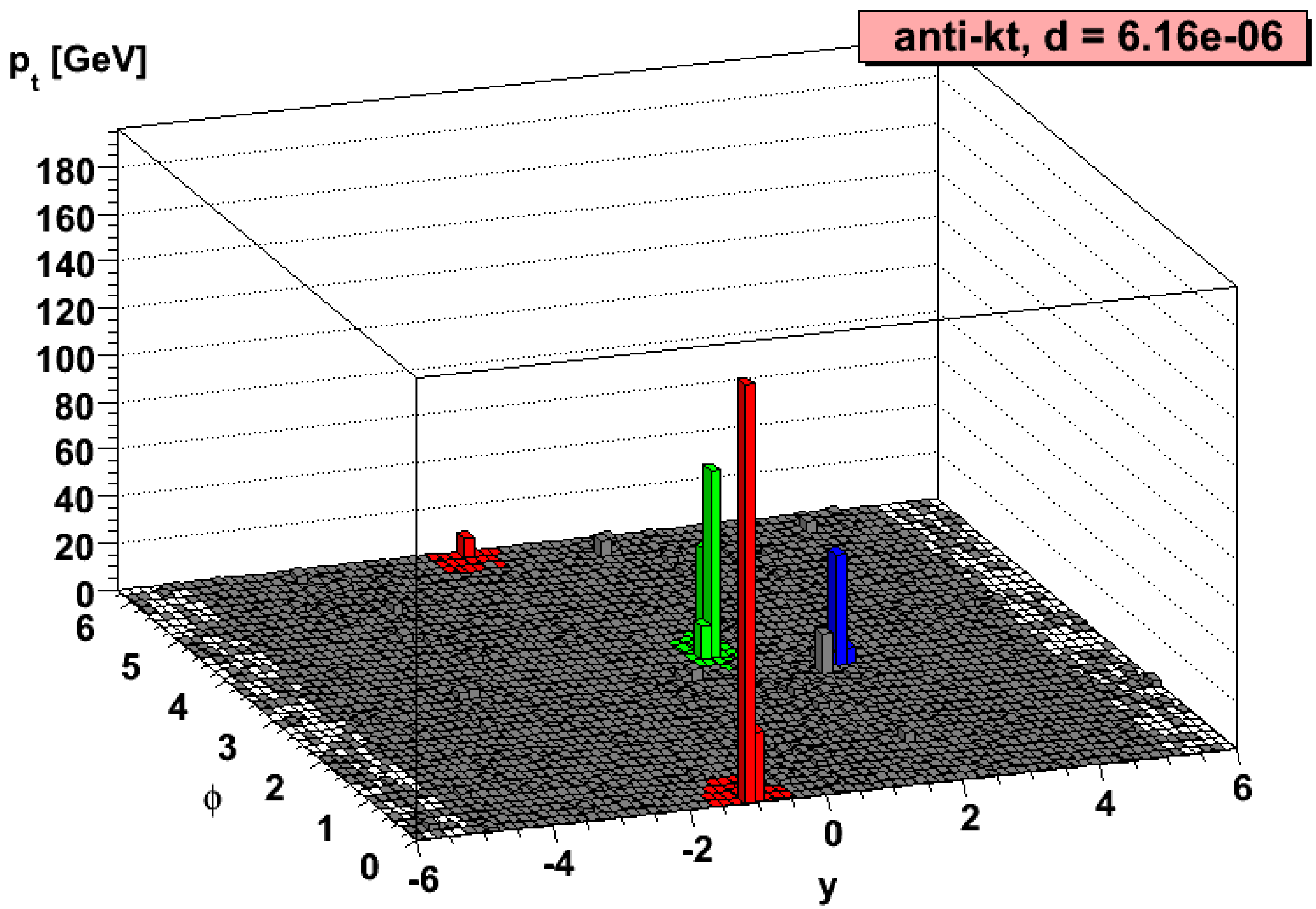
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

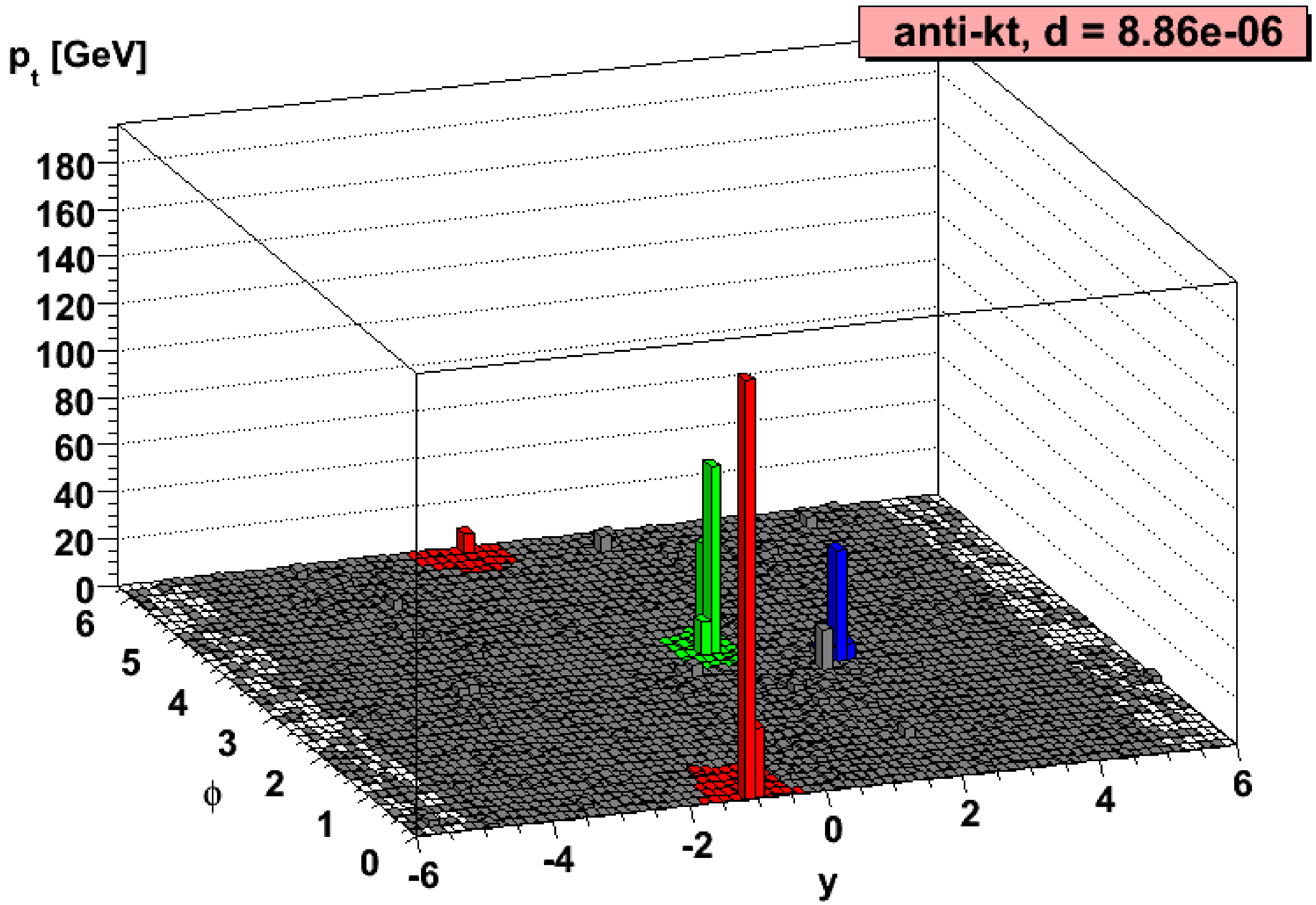
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

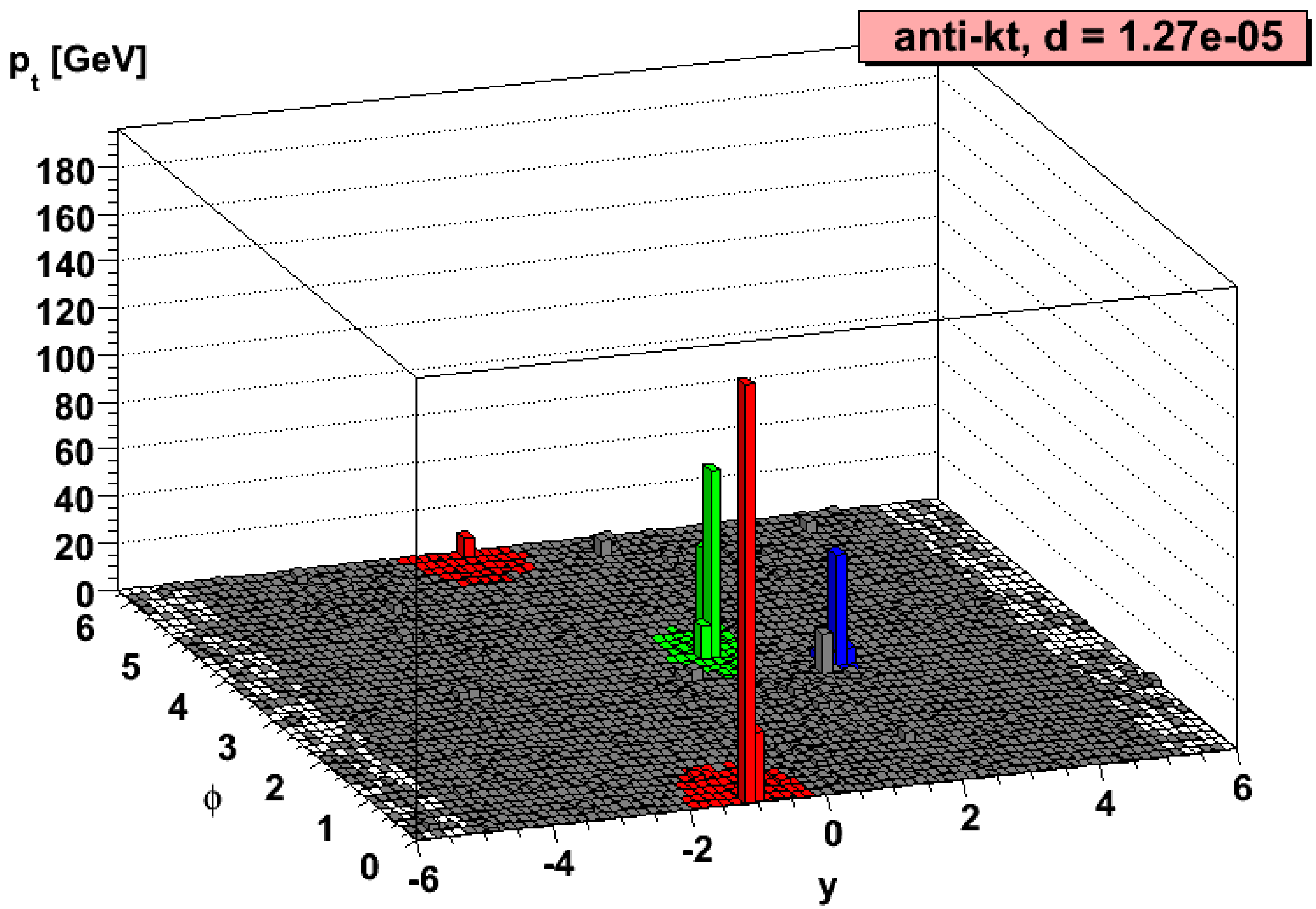
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

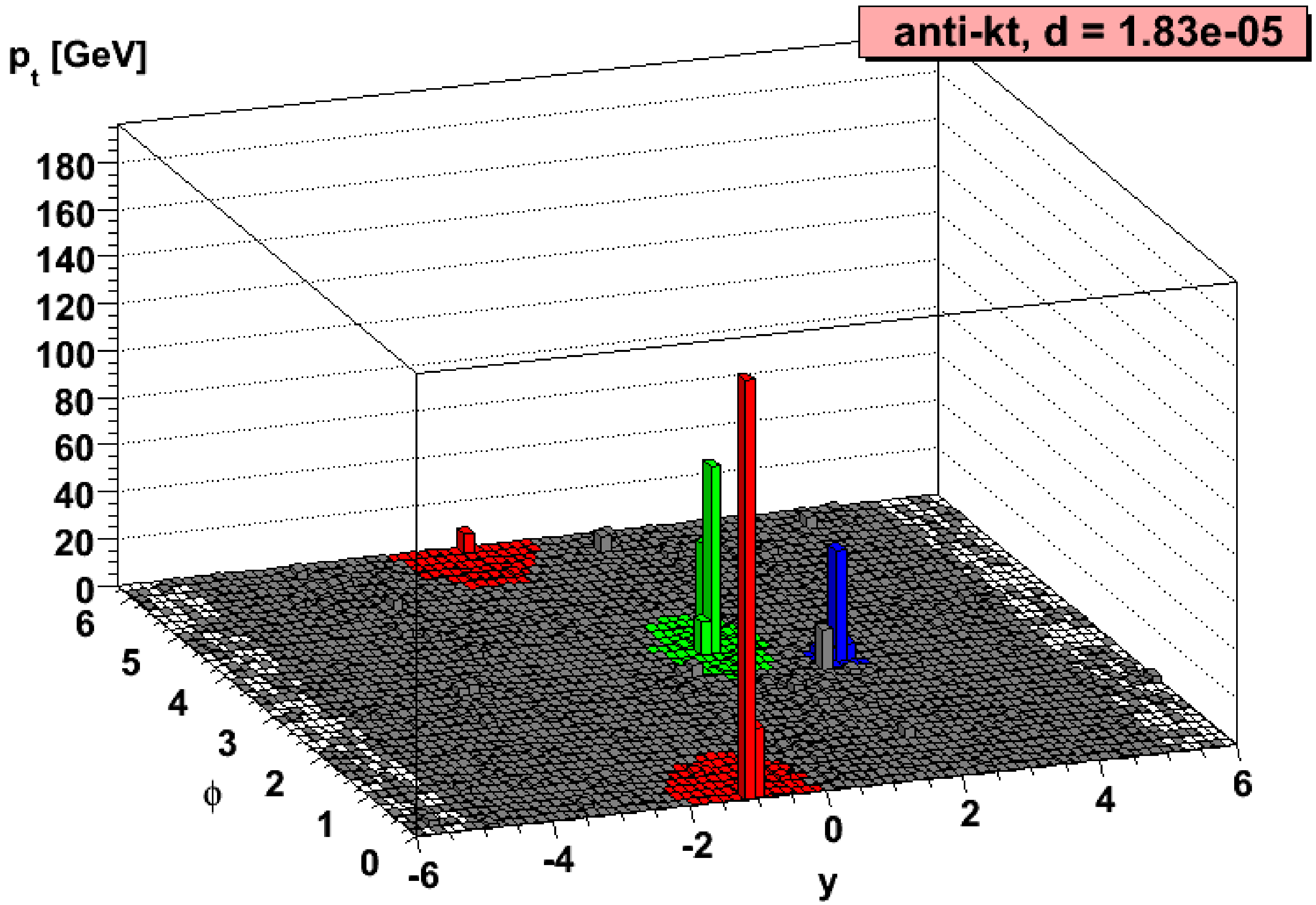
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k$_t$ in action

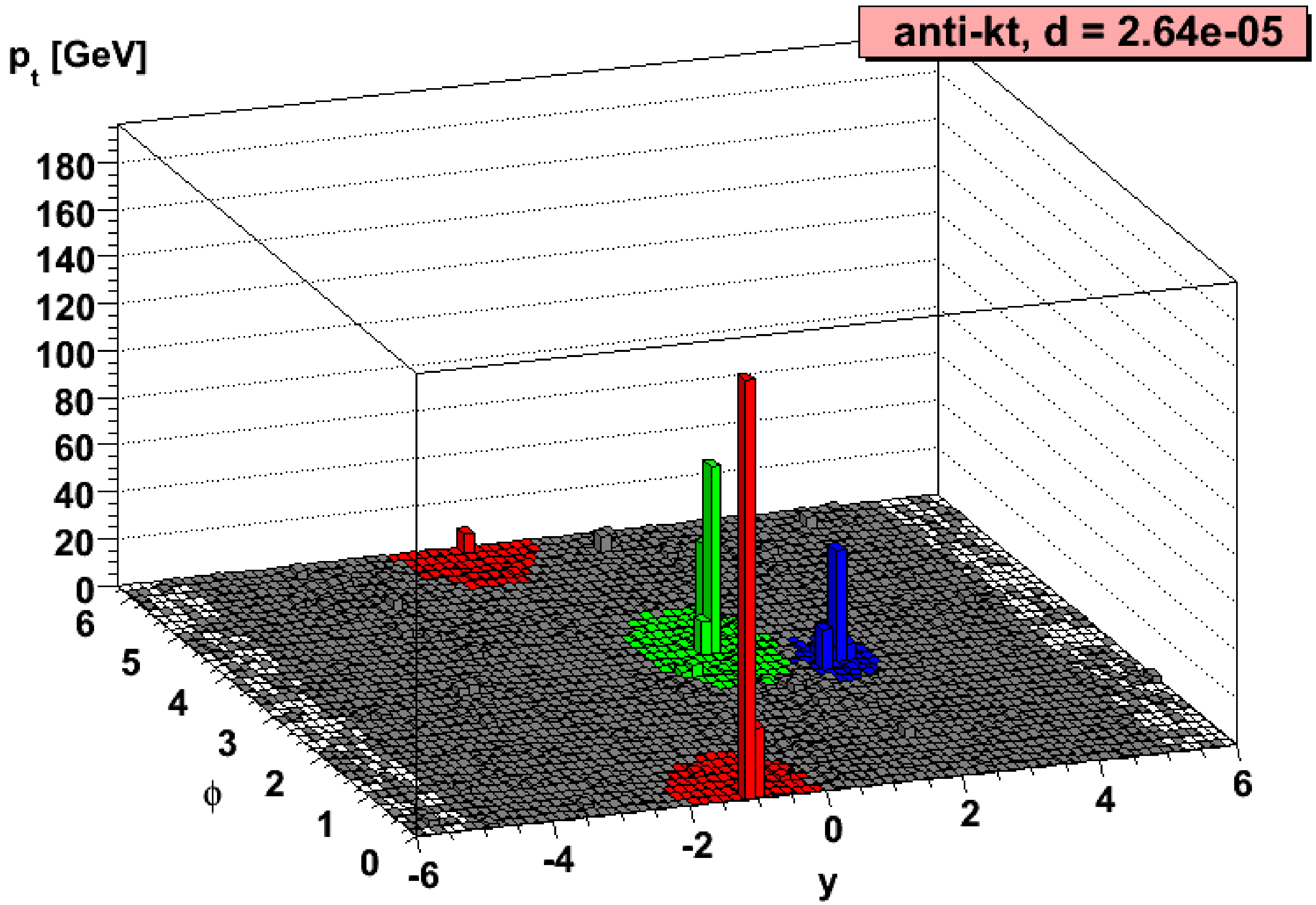
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

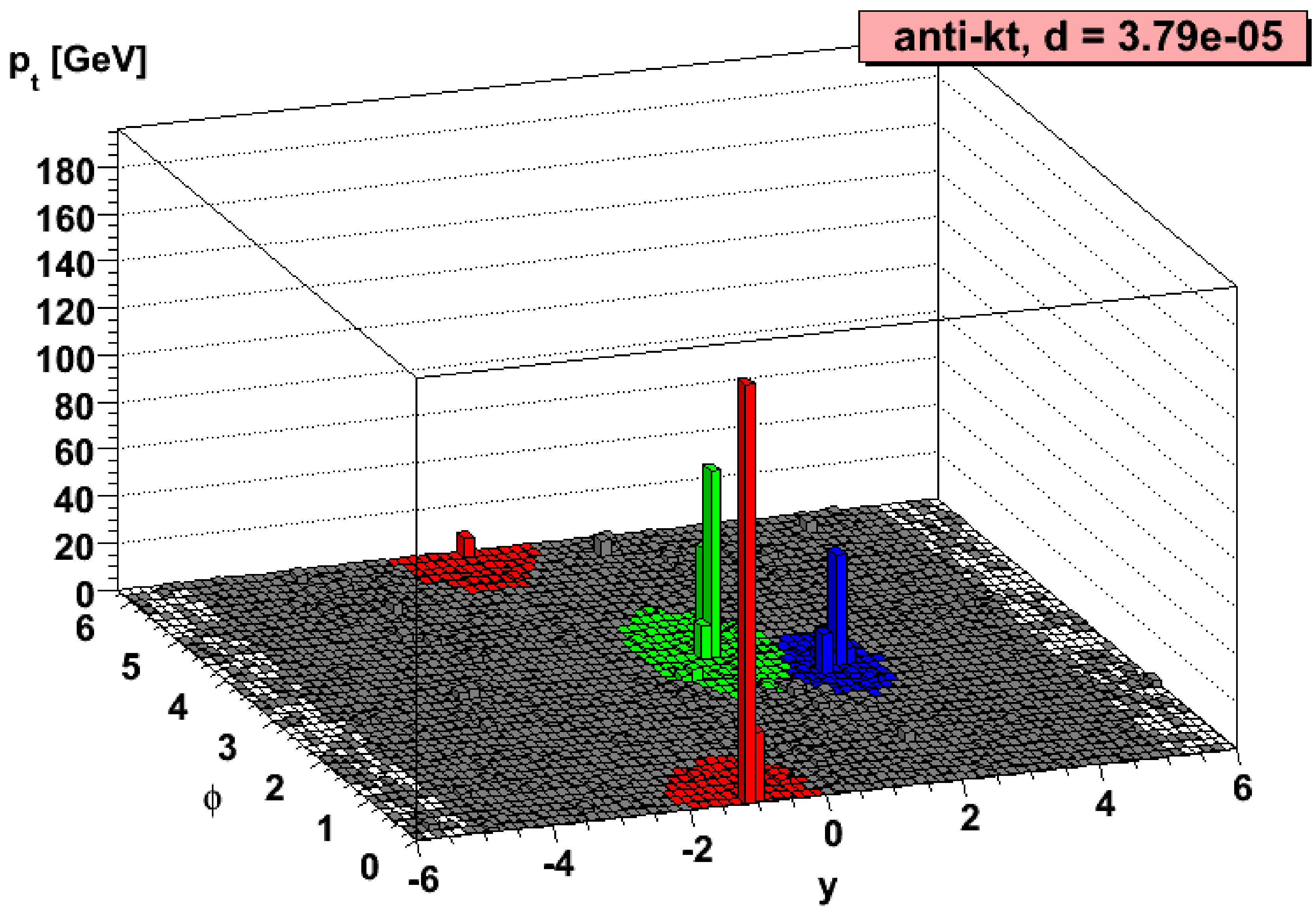
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

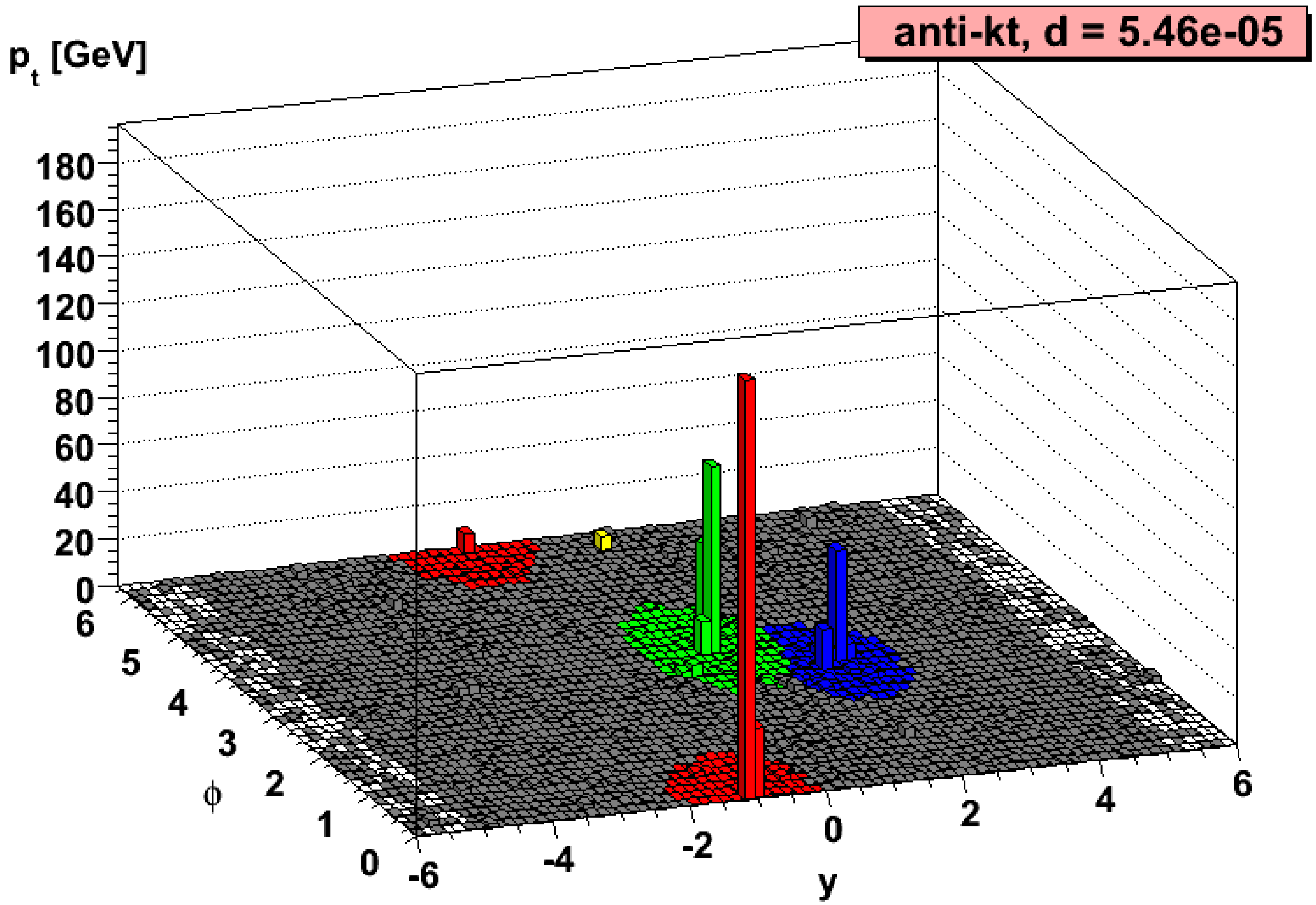
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

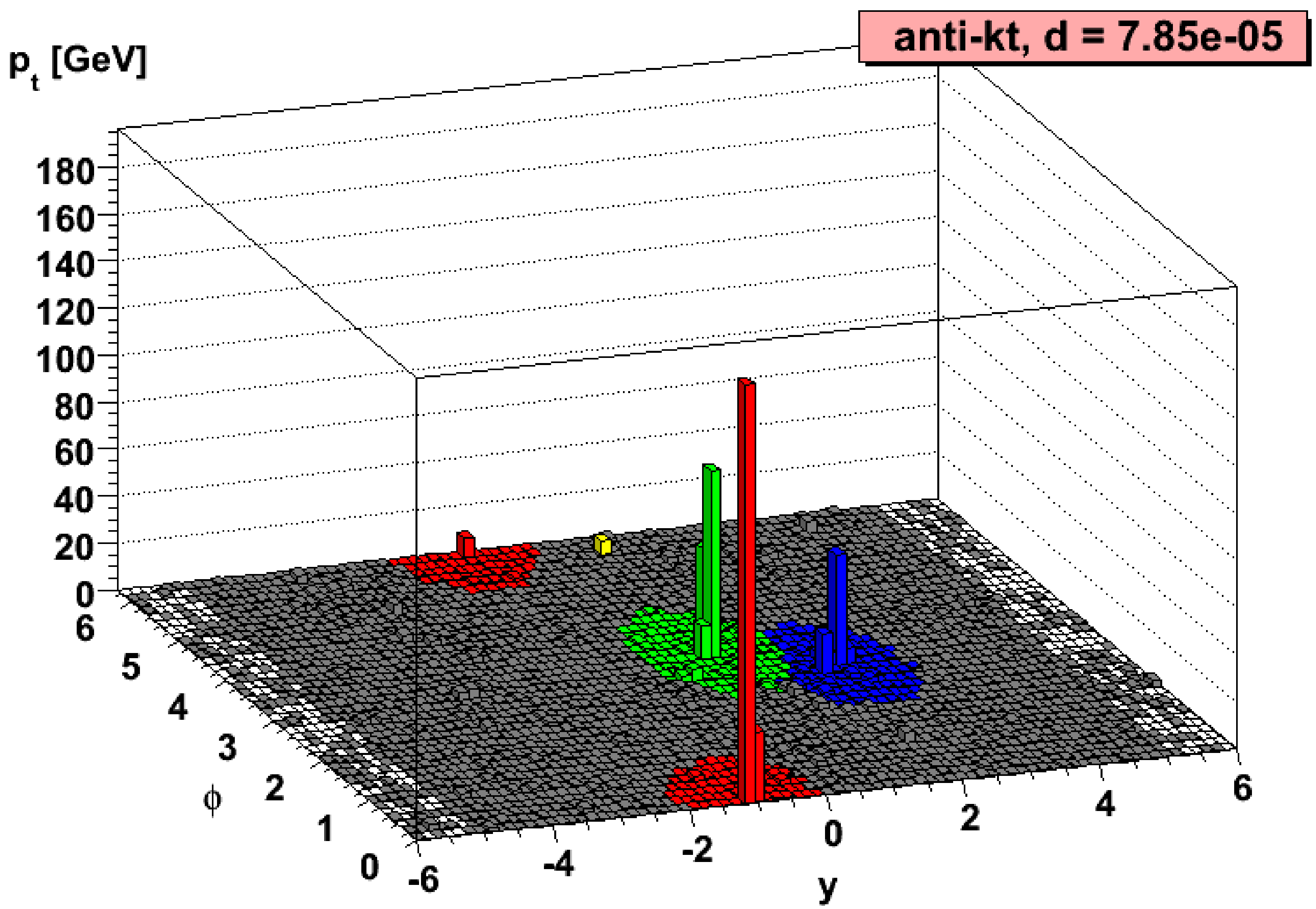
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

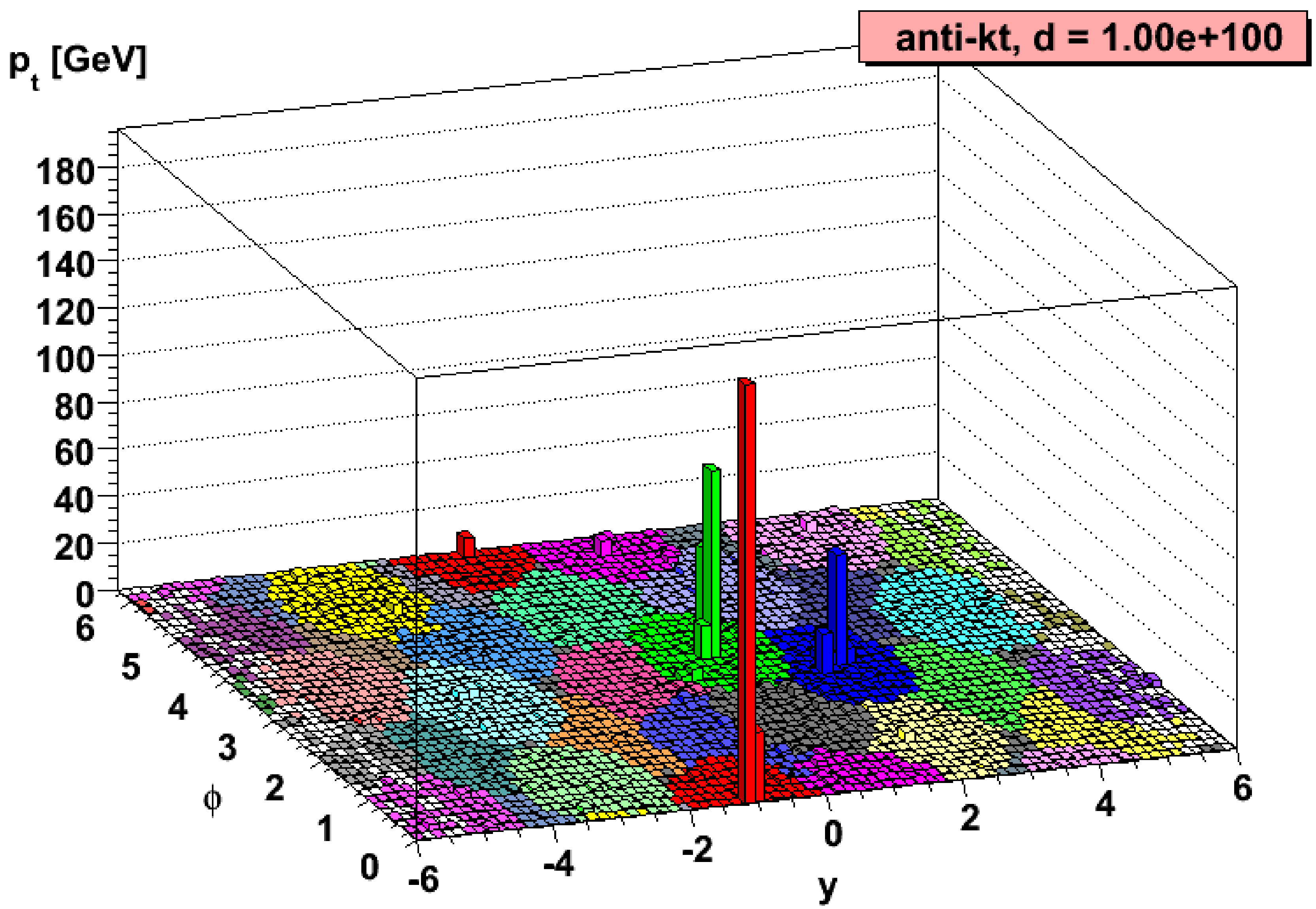
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

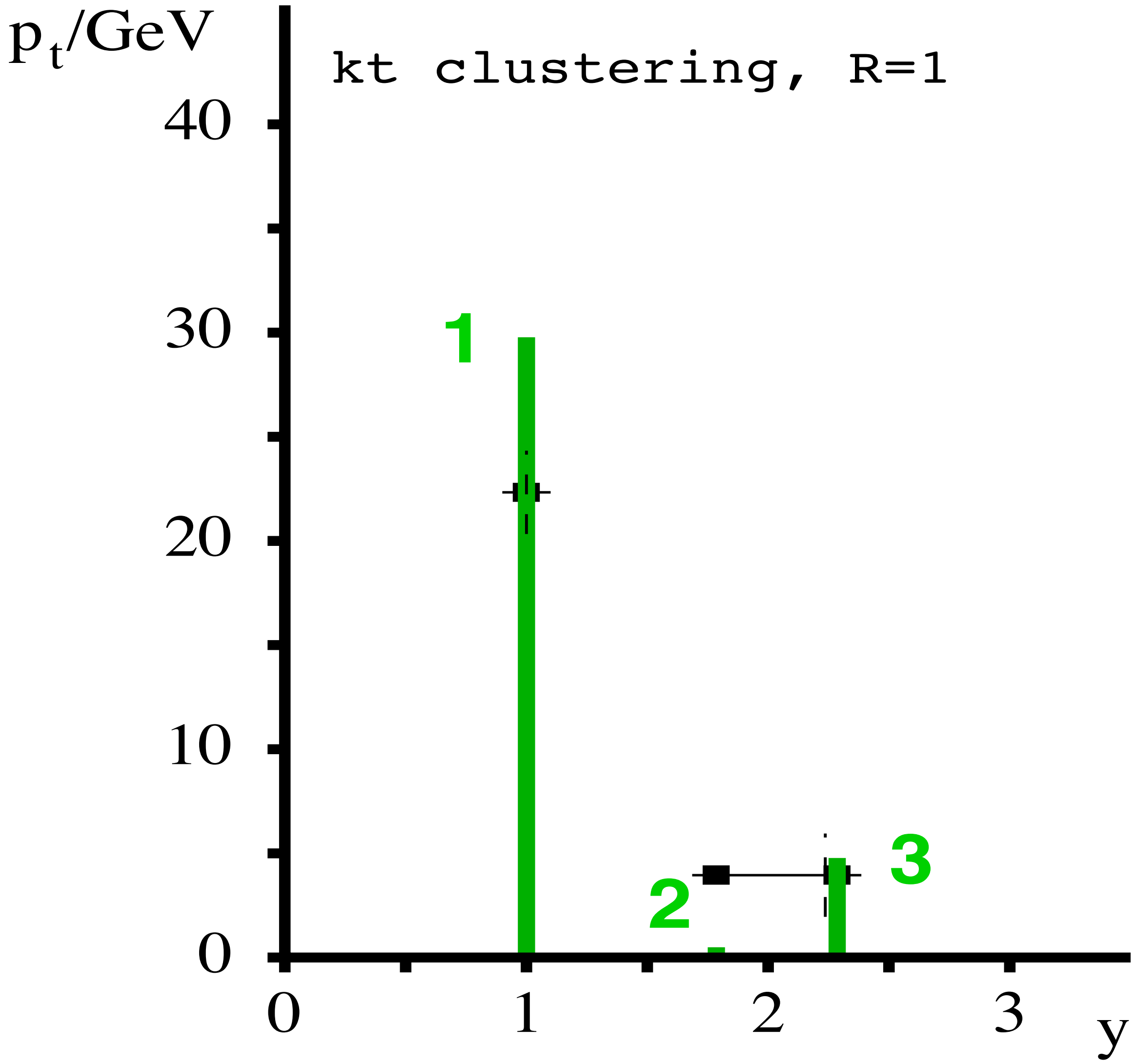
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



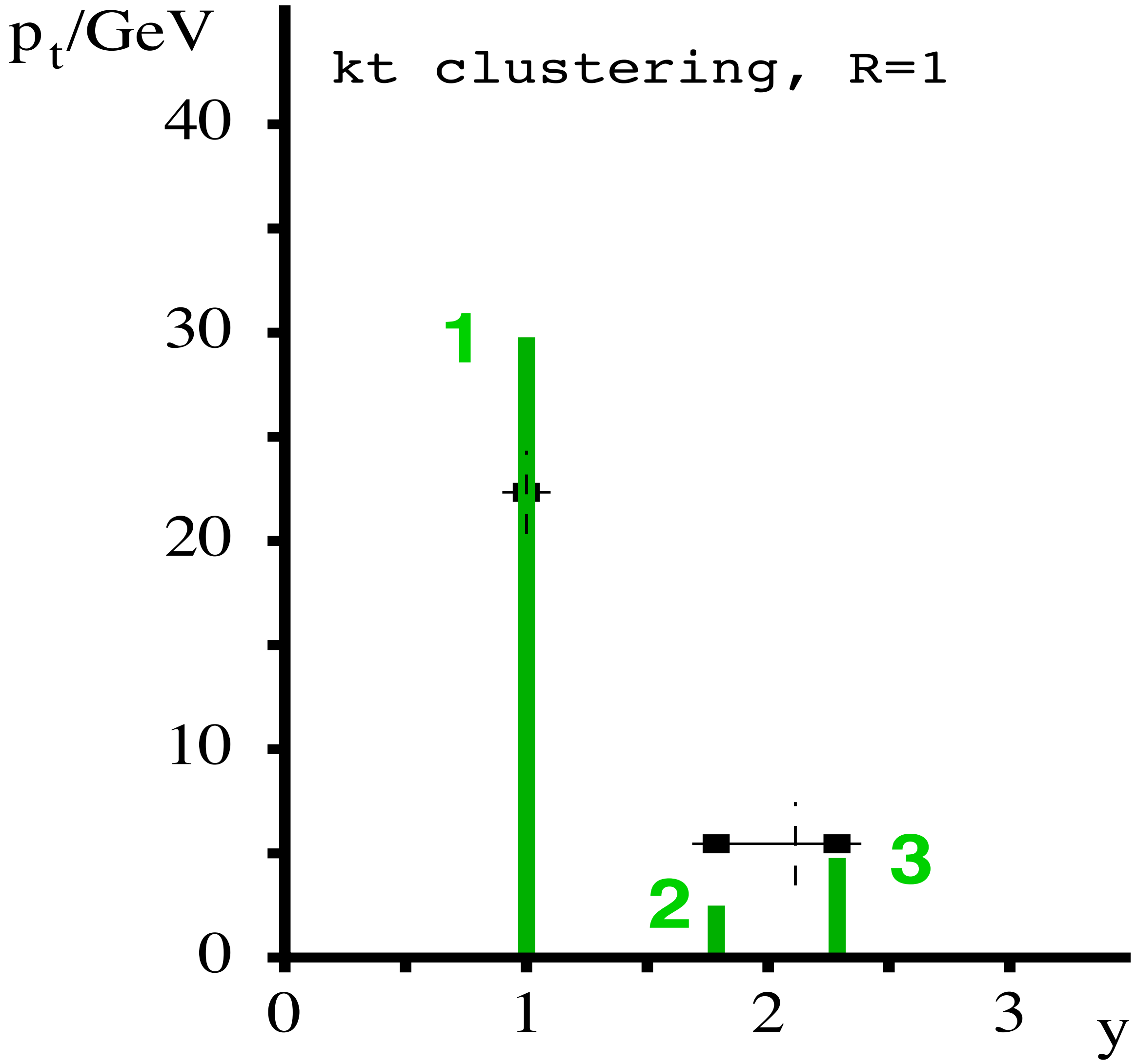
Anti-k_t gives cone-like jets without using stable cones

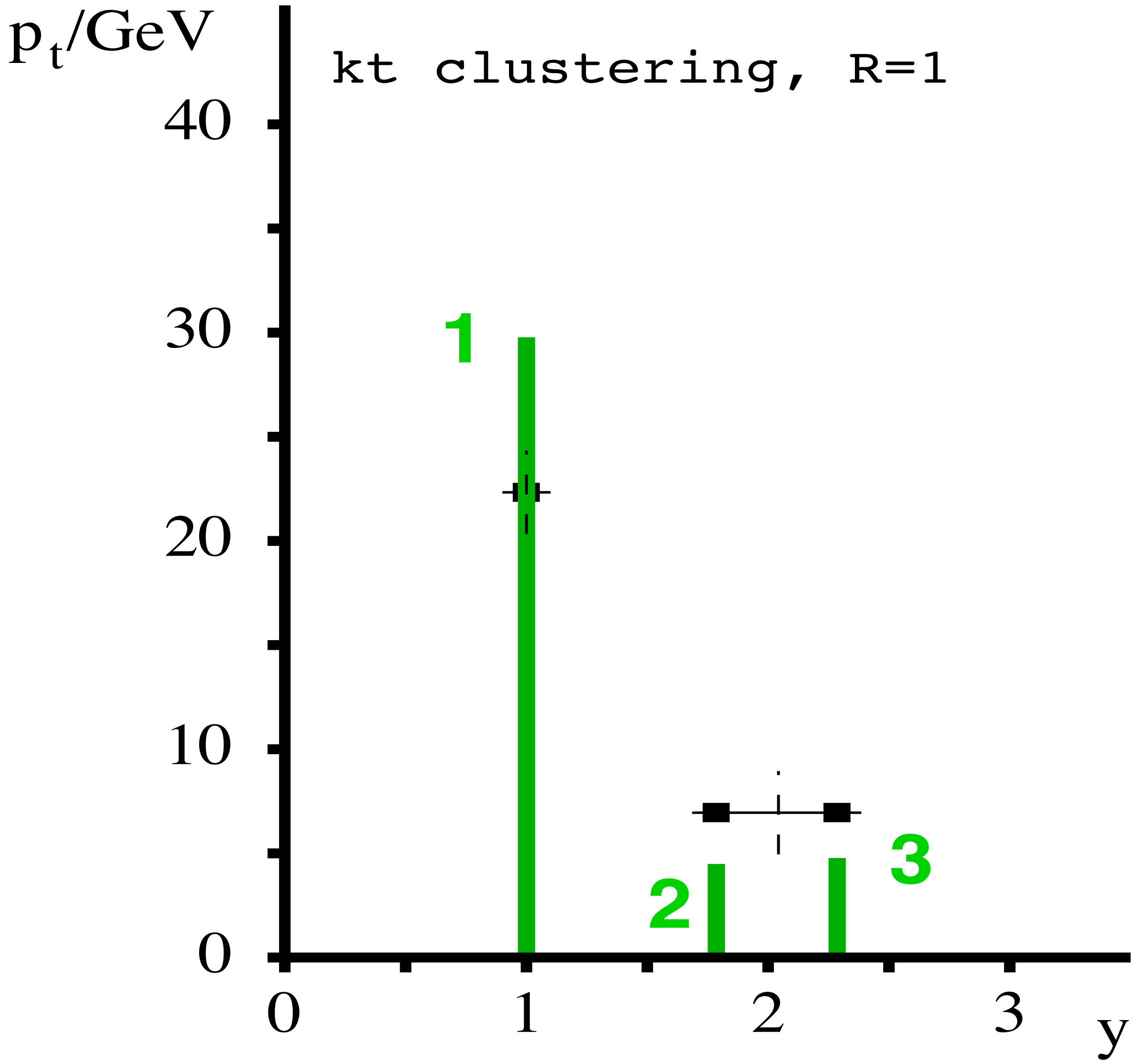
Is it really only about the “circularity” of the jets’ boundaries?

Linearity: k_t v. anti-k_t



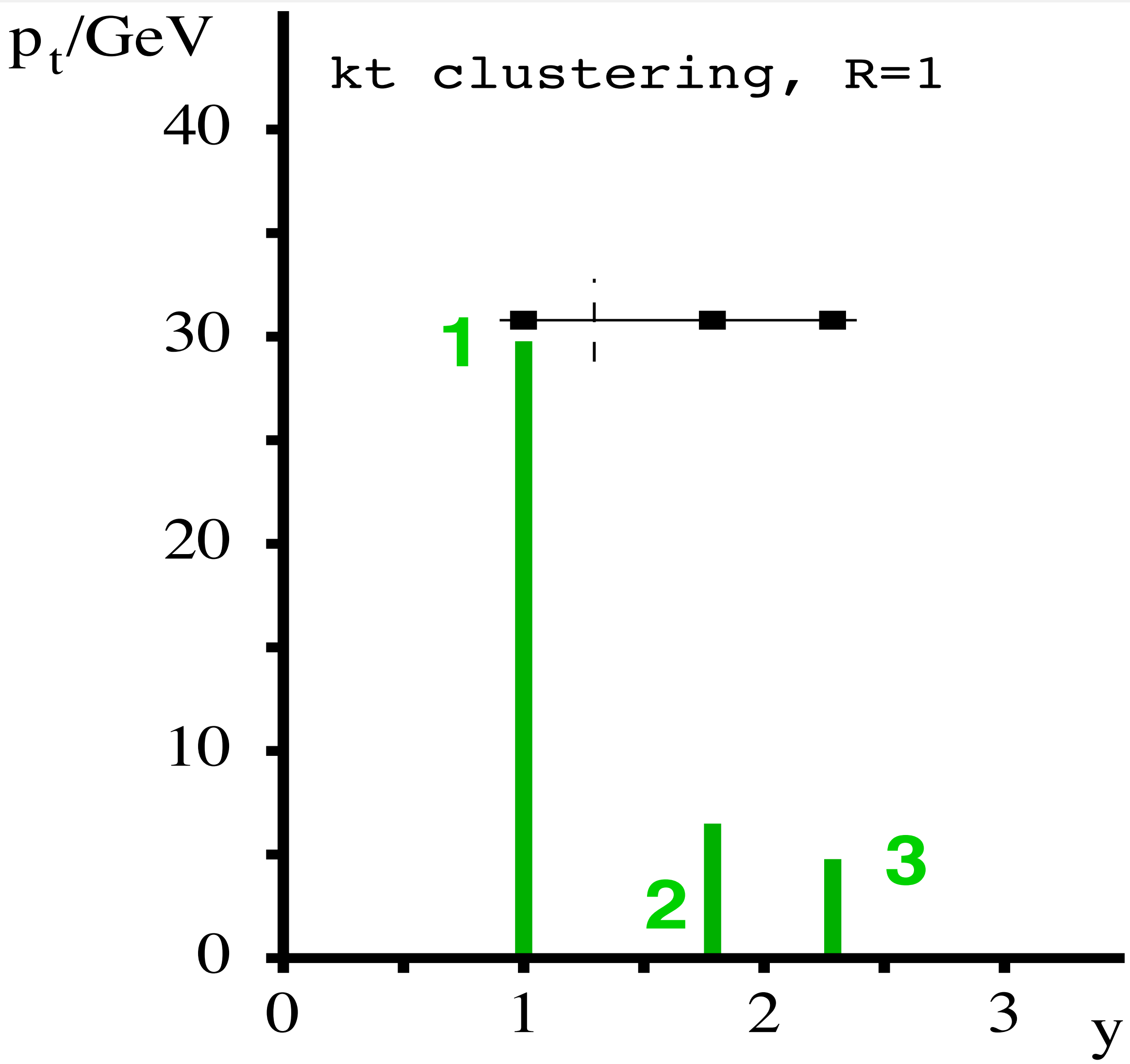
Linearity: k_t v. anti-k_t

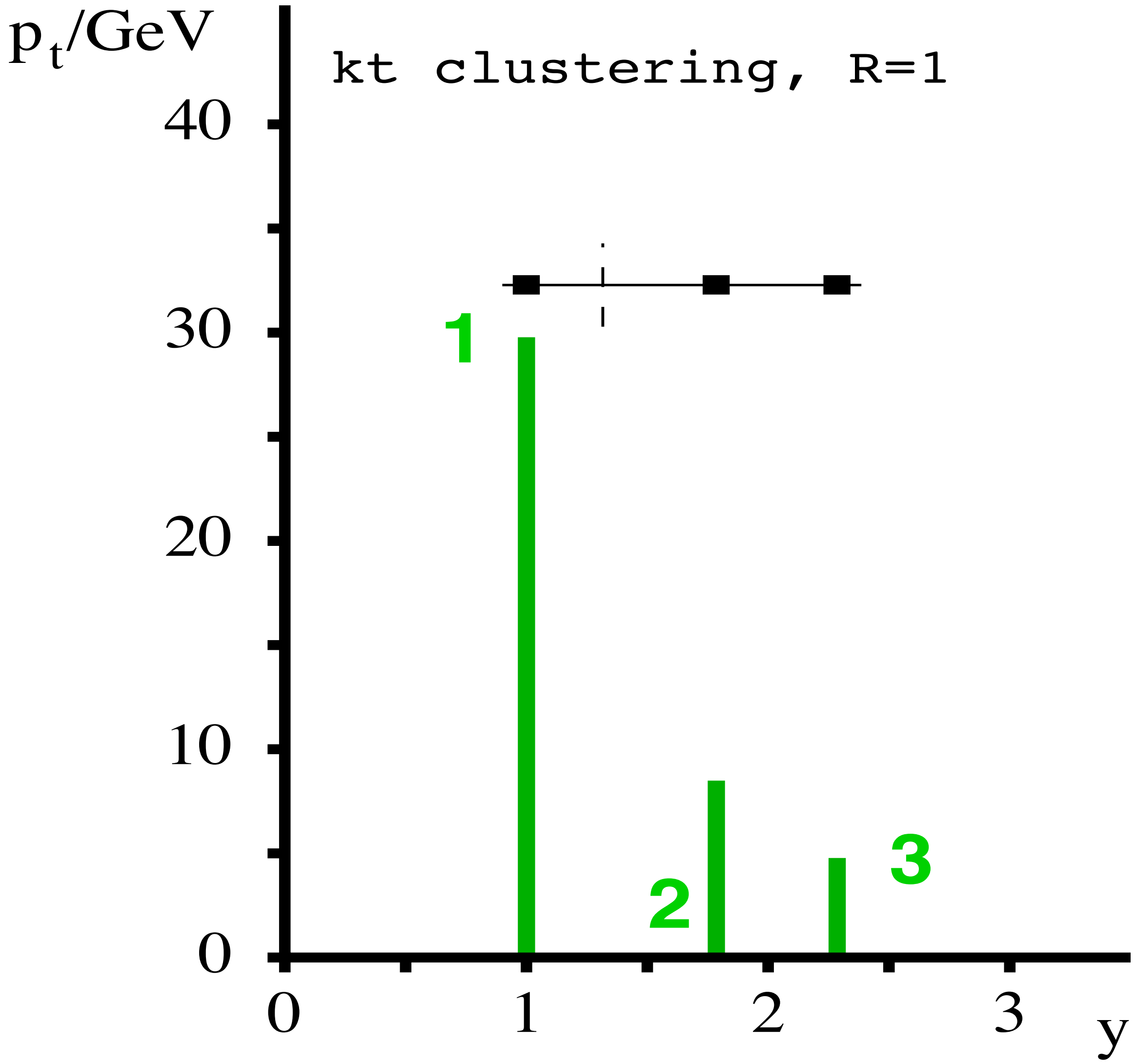




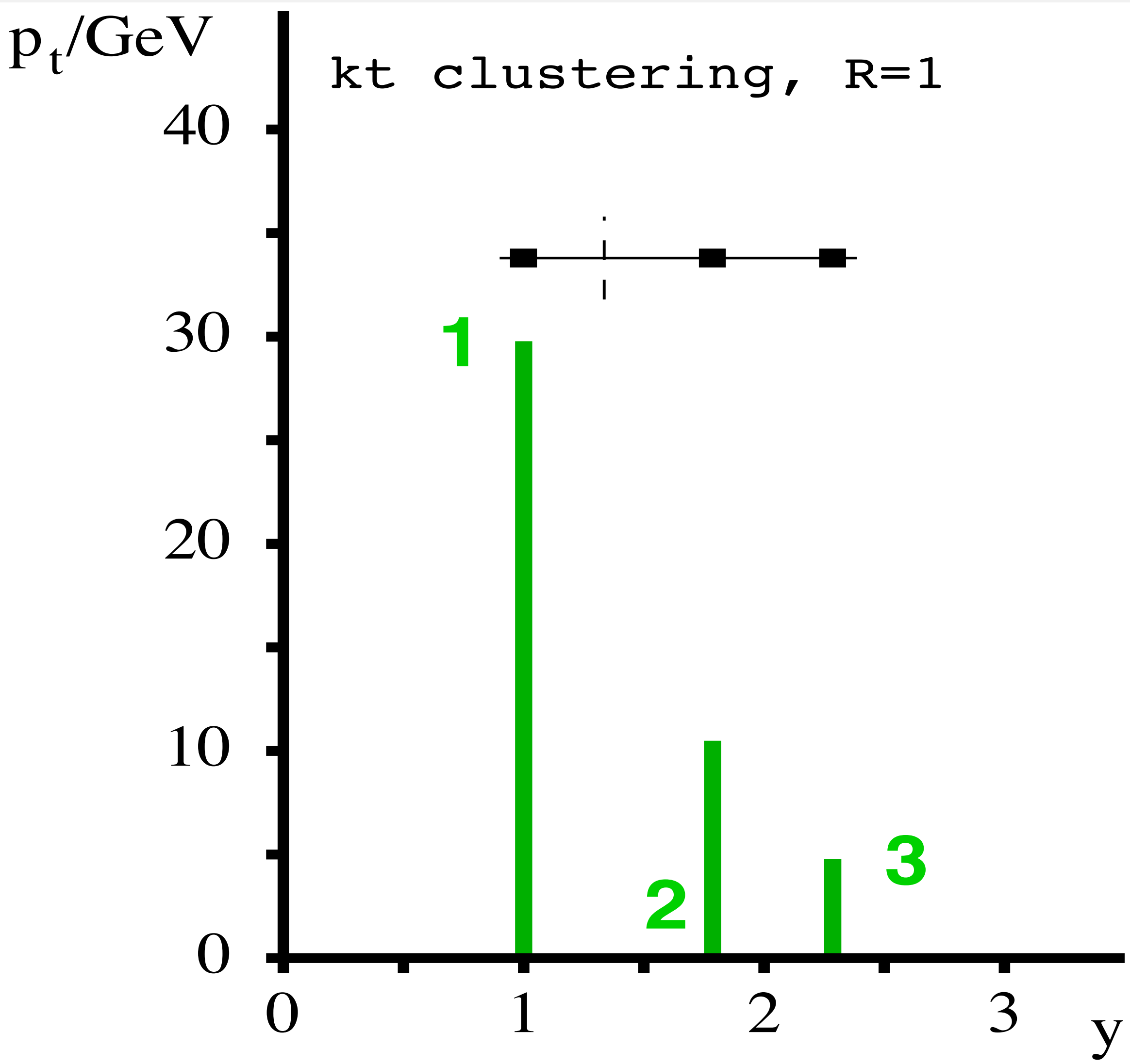
p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

Linearity: k_t v. anti-k_t

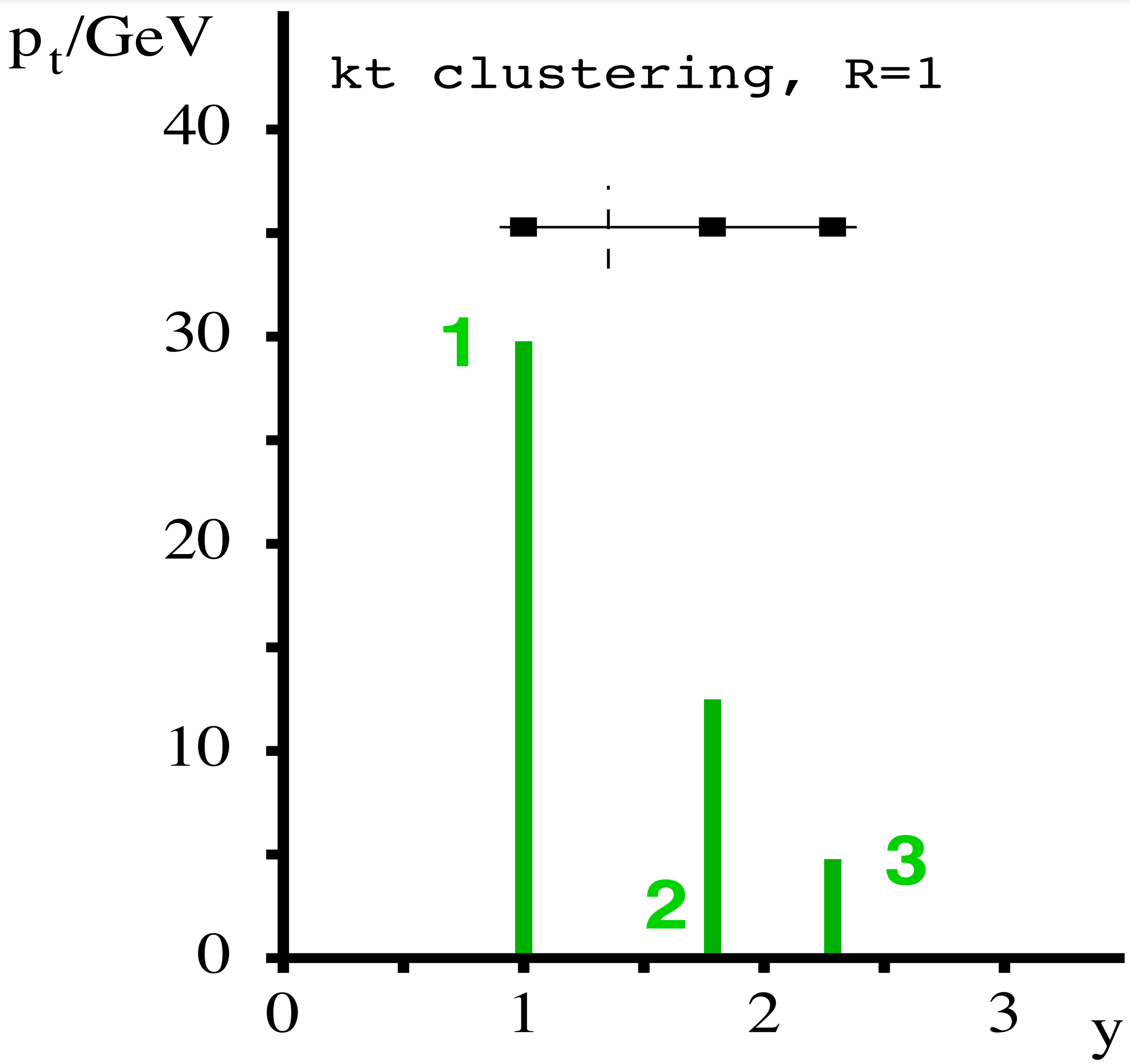


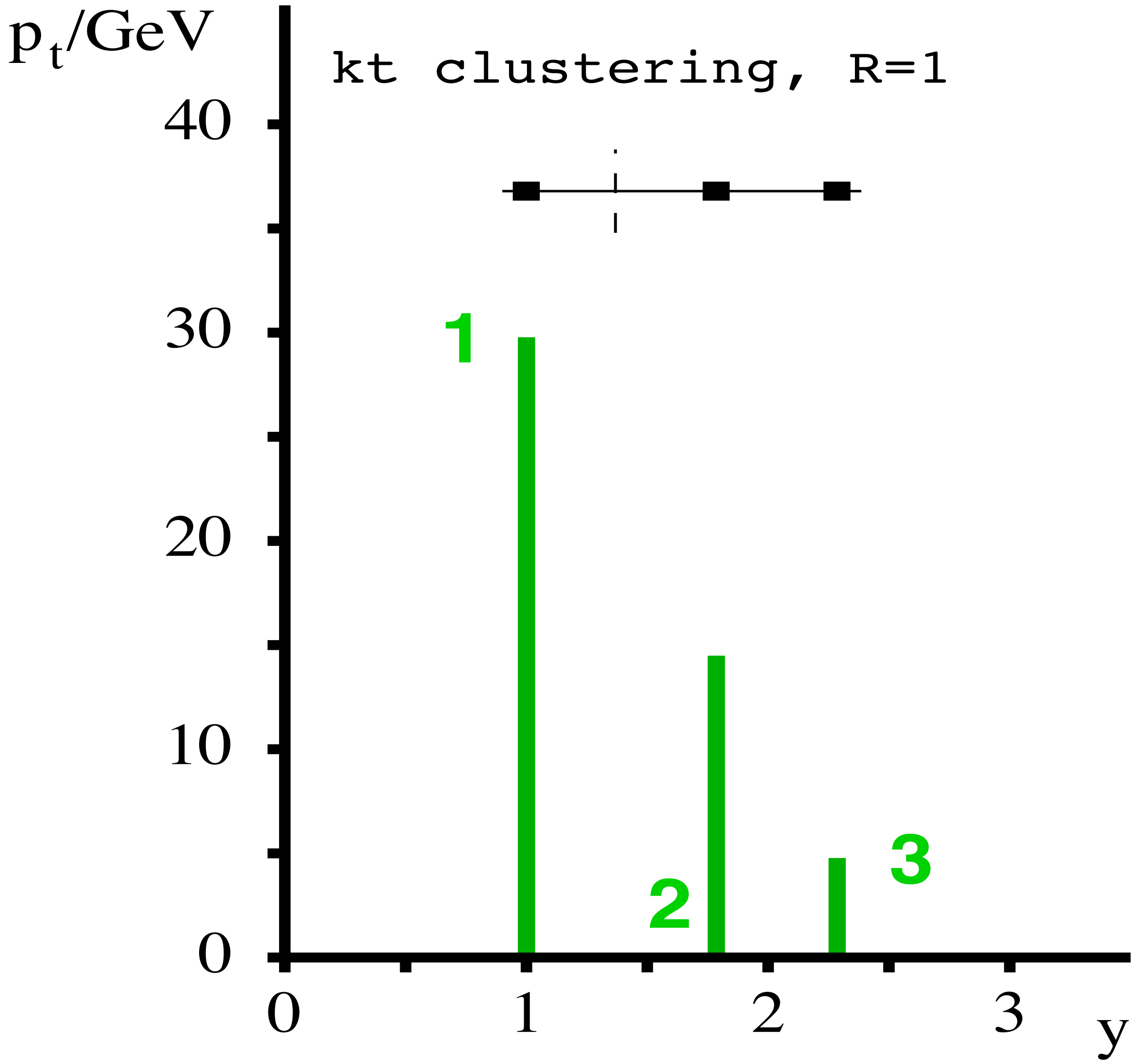

p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

Linearity: k_t v. anti-k_t

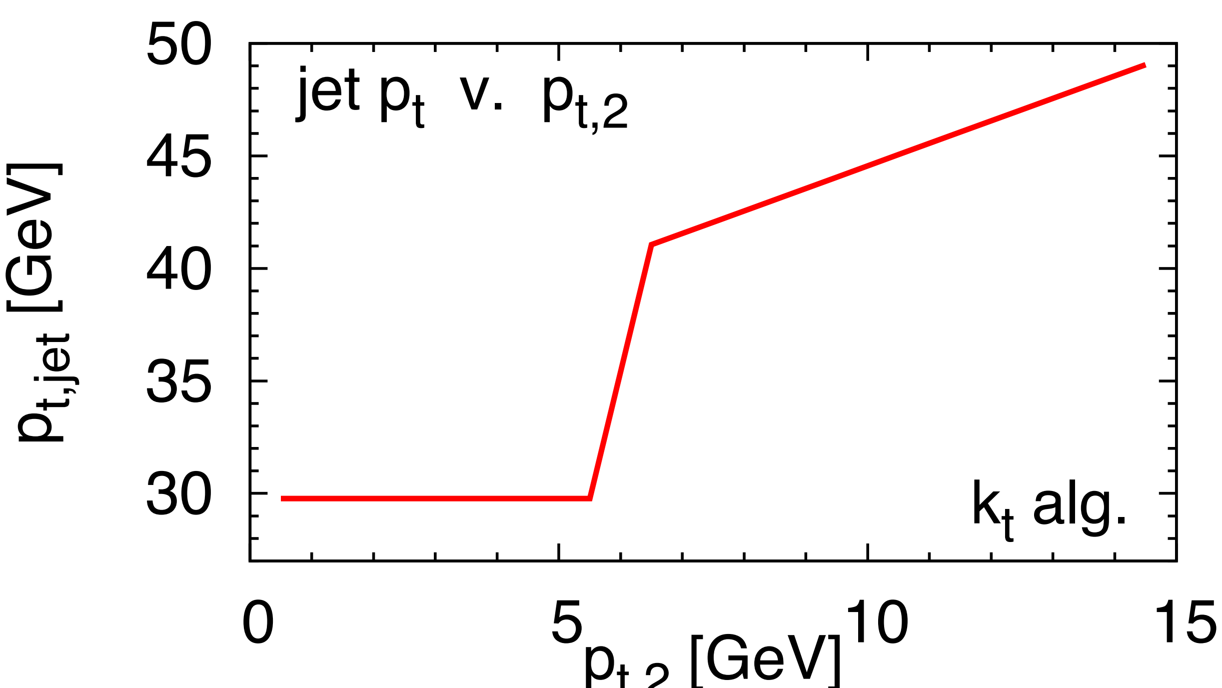


Linearity: k_t v. anti-k_t

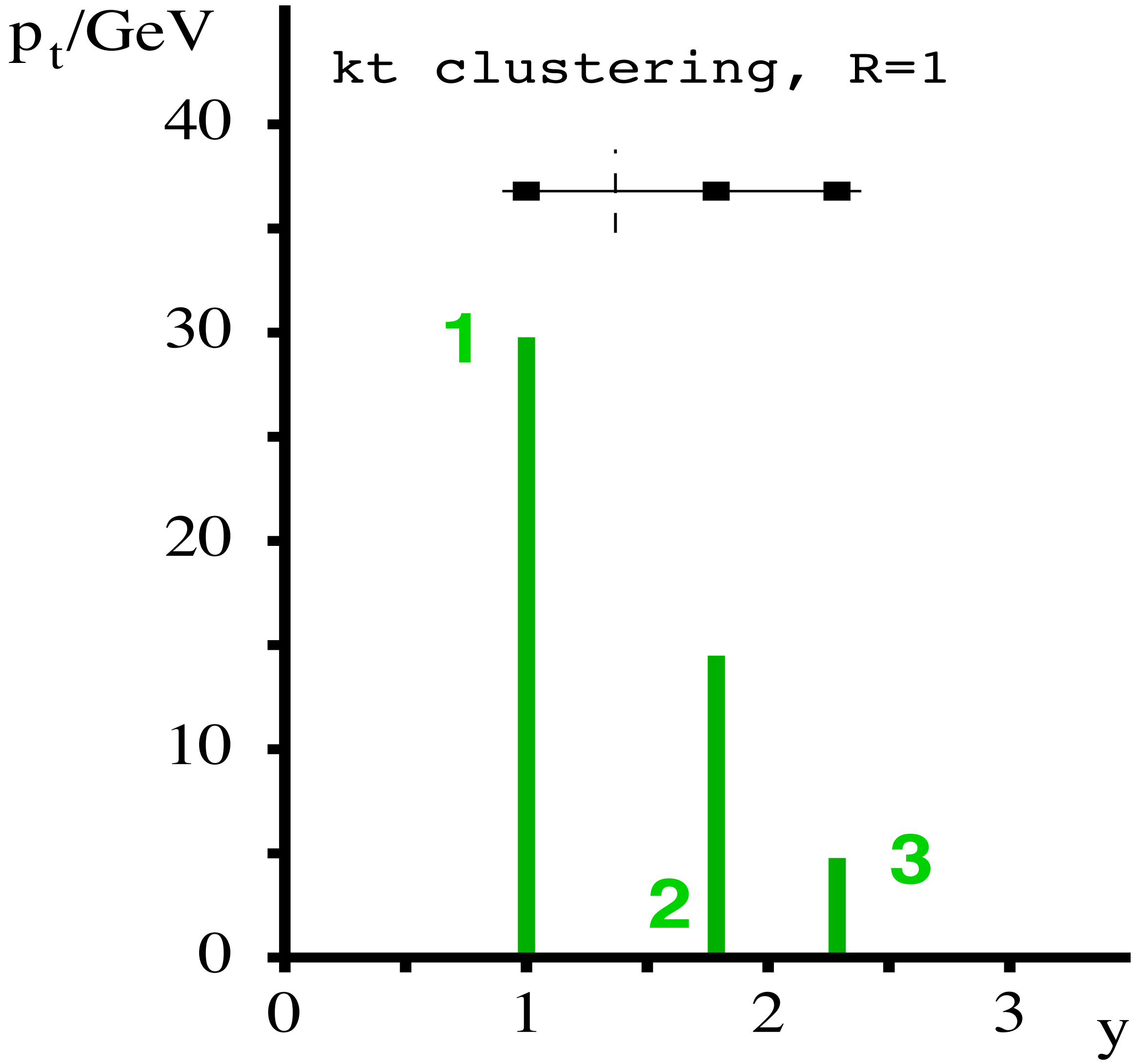


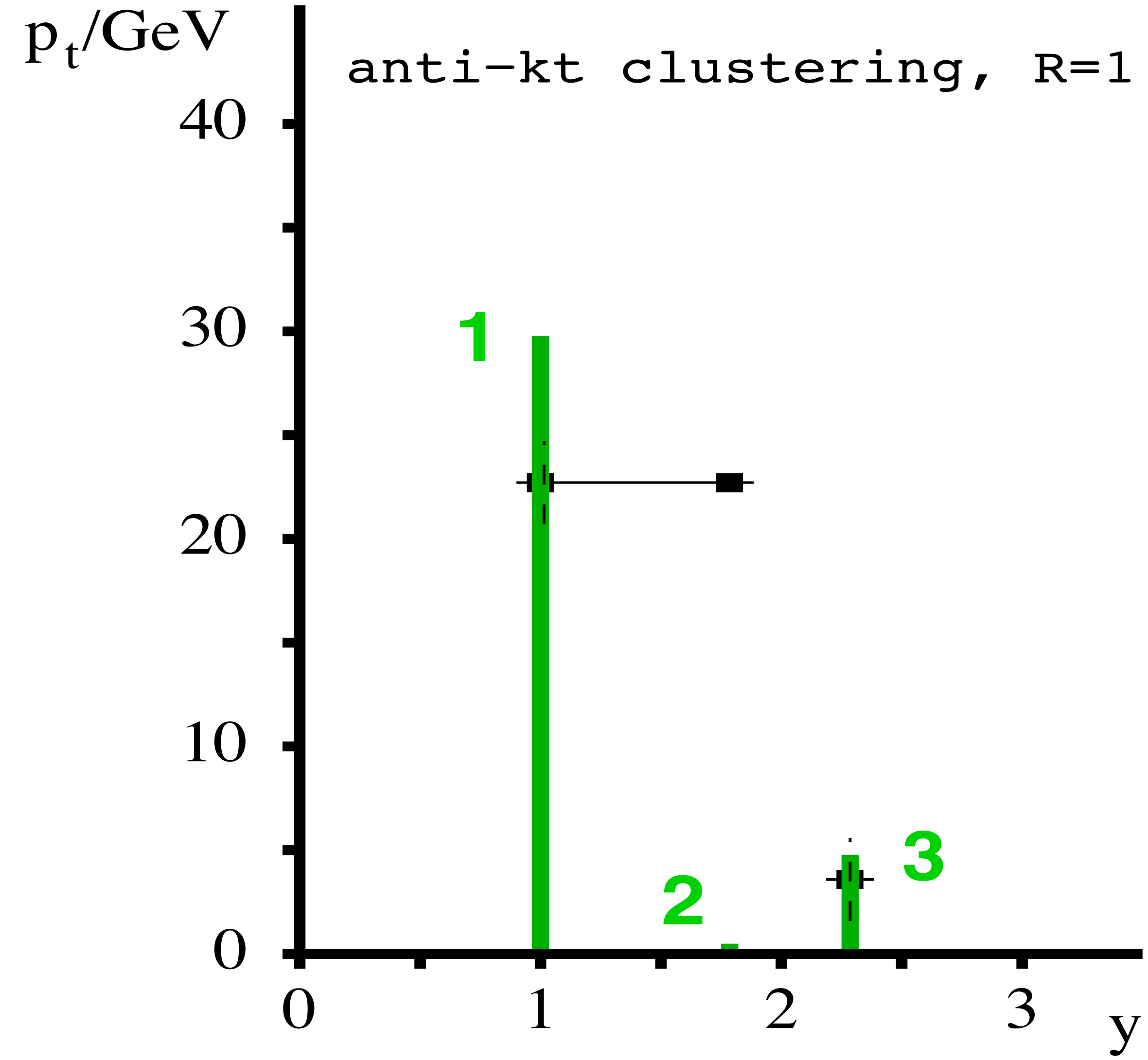


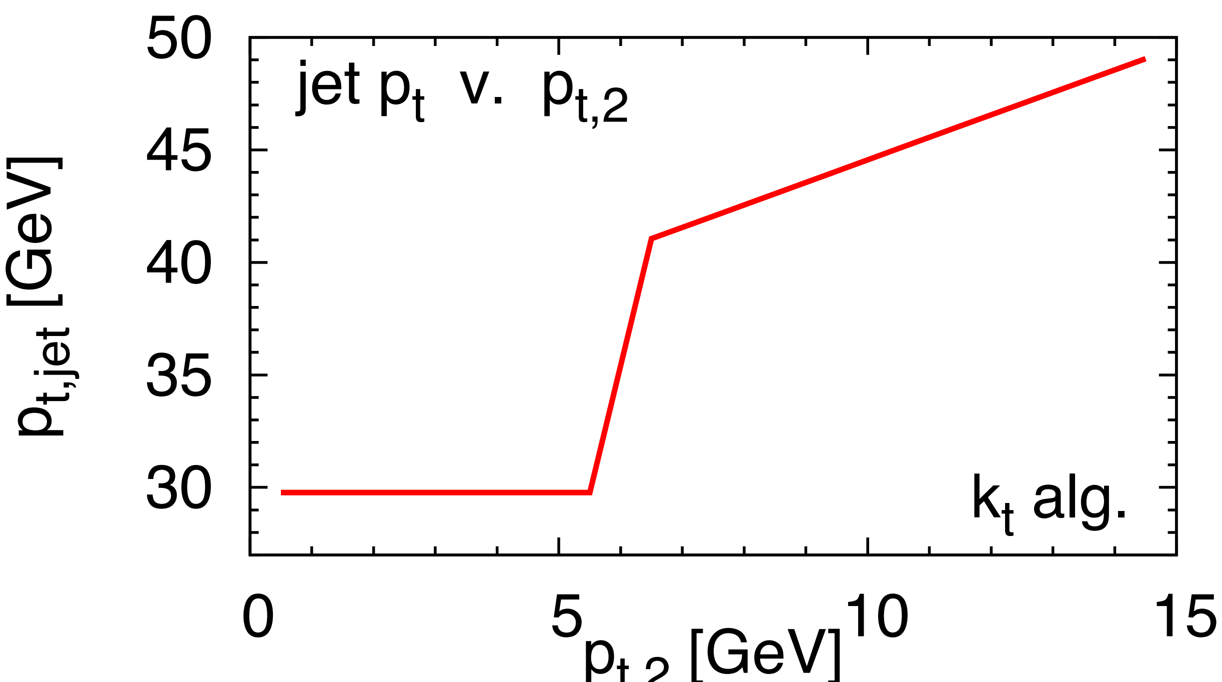
p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y



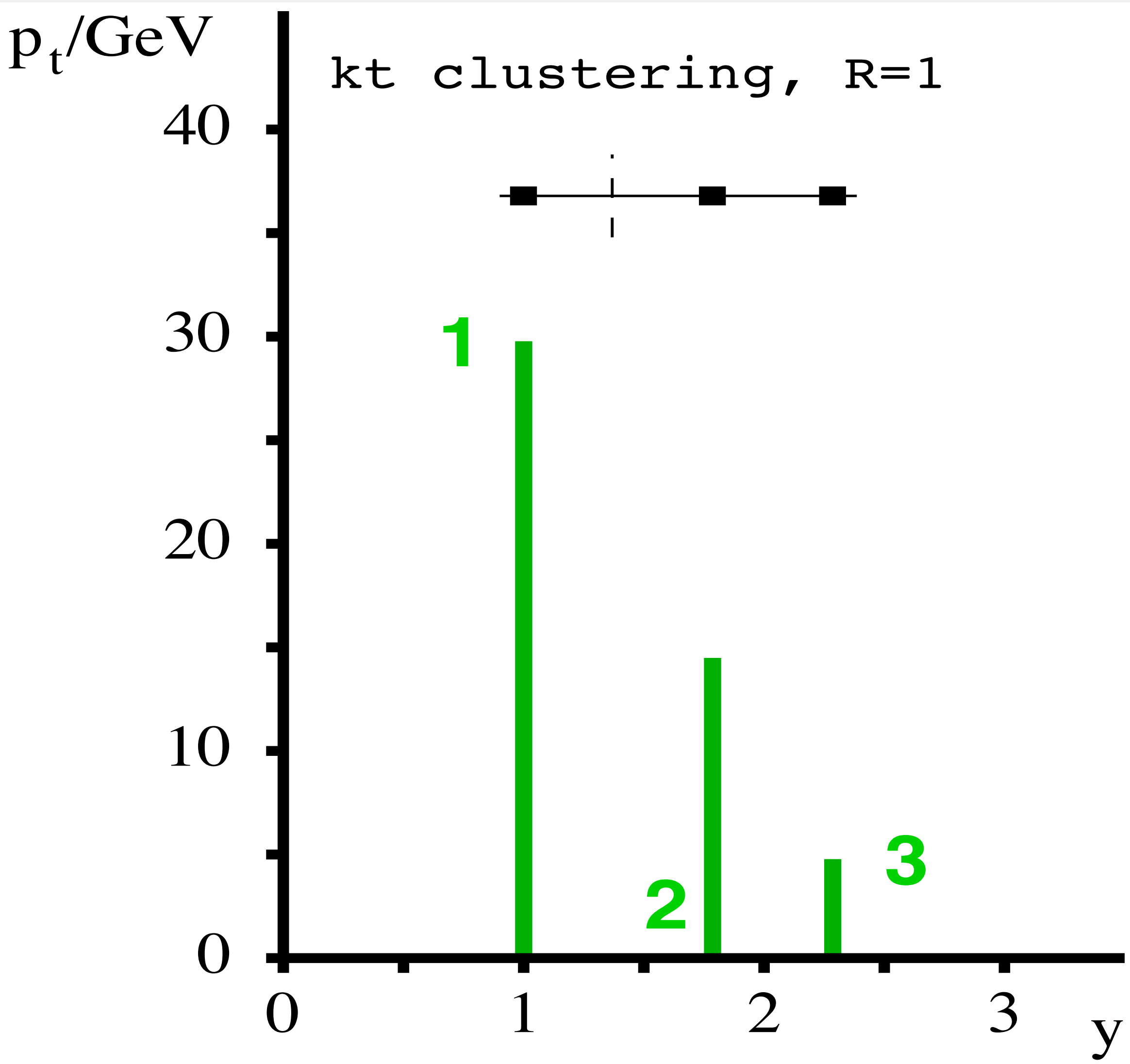
jet p_t v. p_t,2
p_t,jet [GeV]
50
45
40
35
30
k_t alg.
0
5
10
15
p_t,2 [GeV]

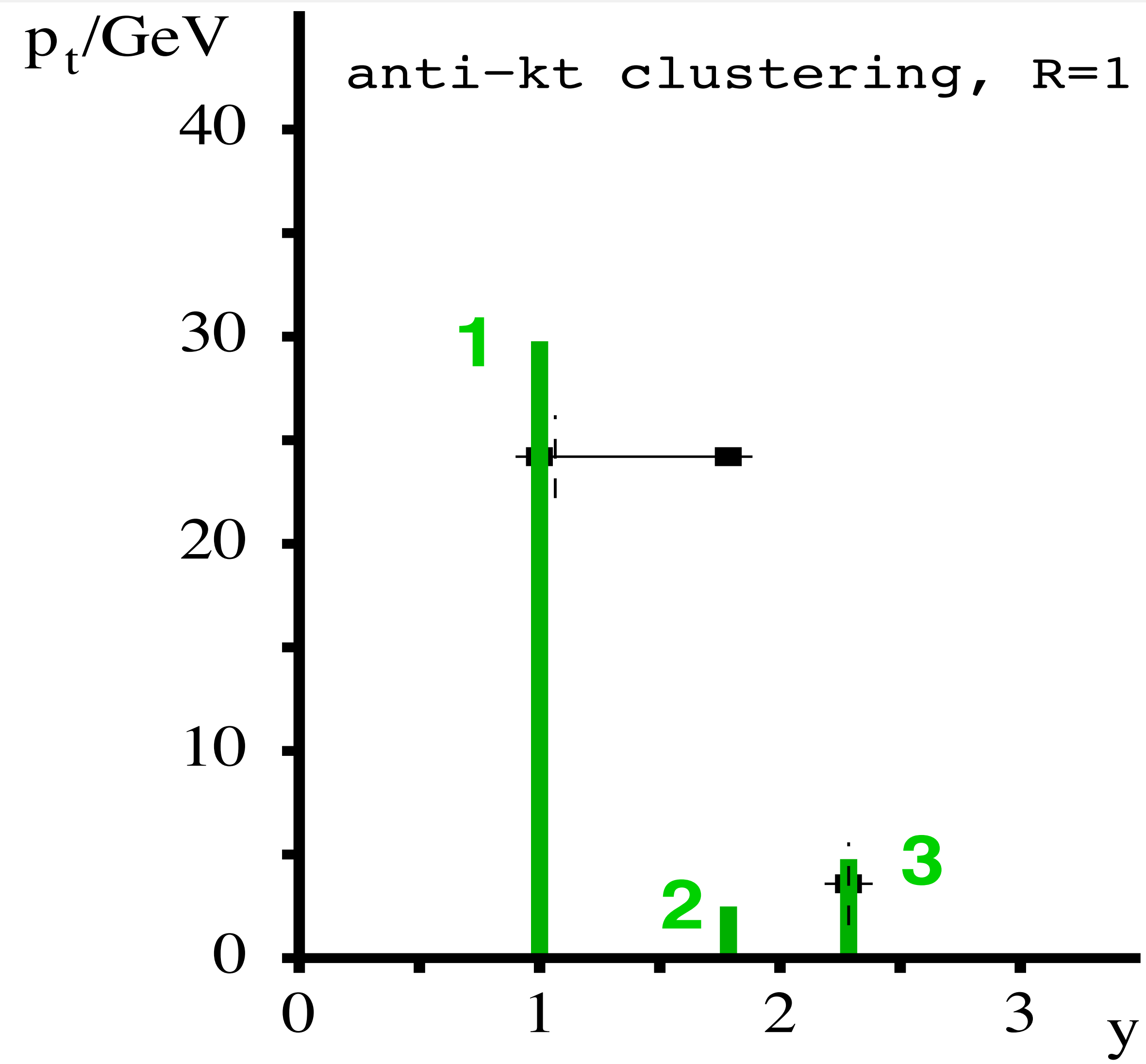

p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

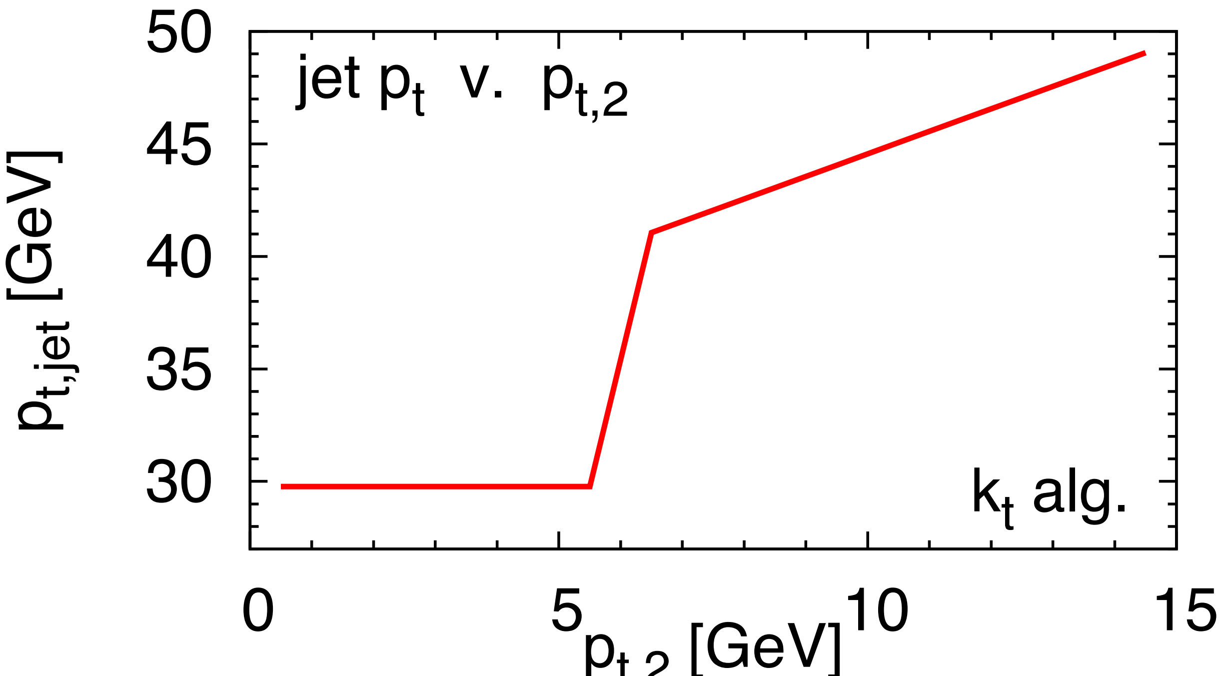

p_t/GeV
anti−kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y


jet p_t v. p_t,2
k_t alg.
p_t,jet [GeV]
50
45
40
35
30
0
5
10
15
p_t,2 [GeV]

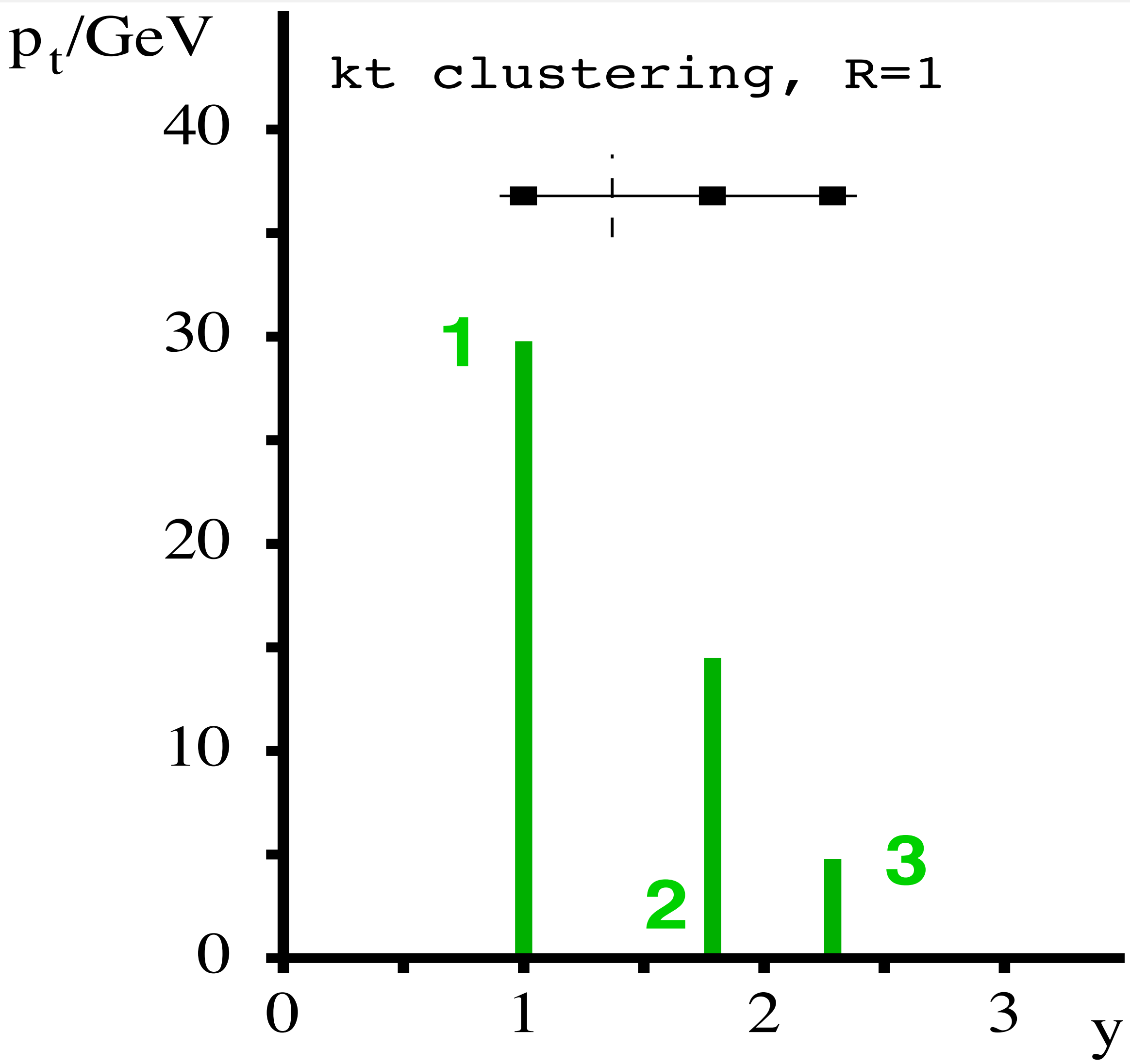
Linearity: k_t v. anti-k_t

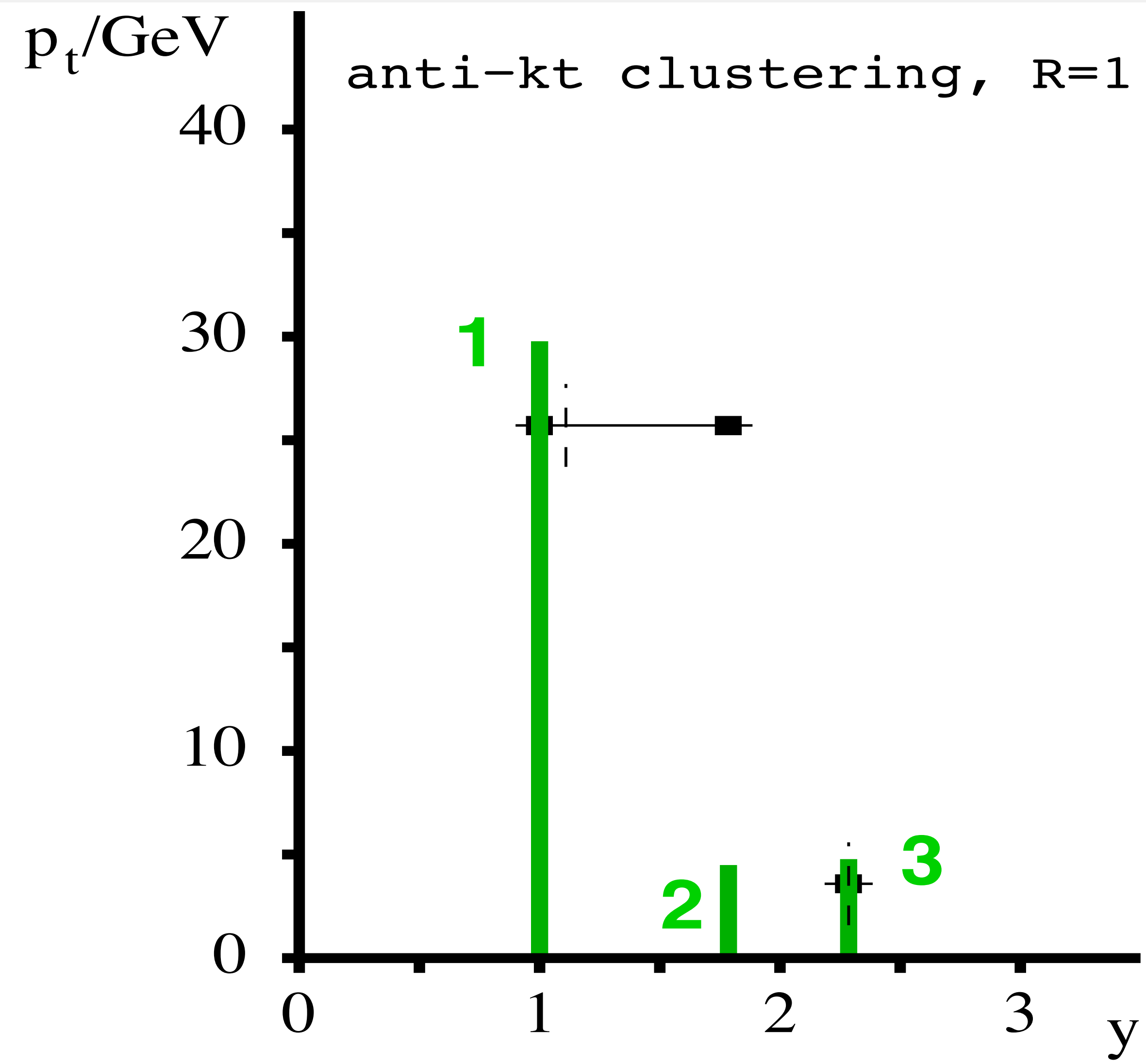


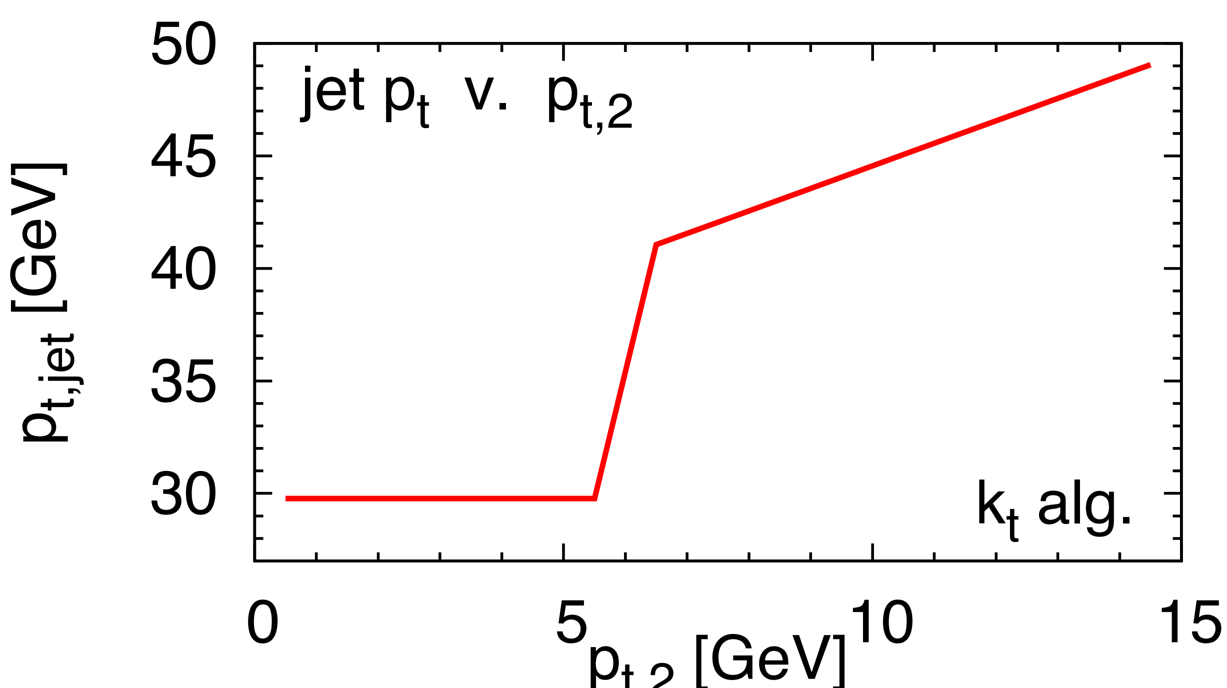


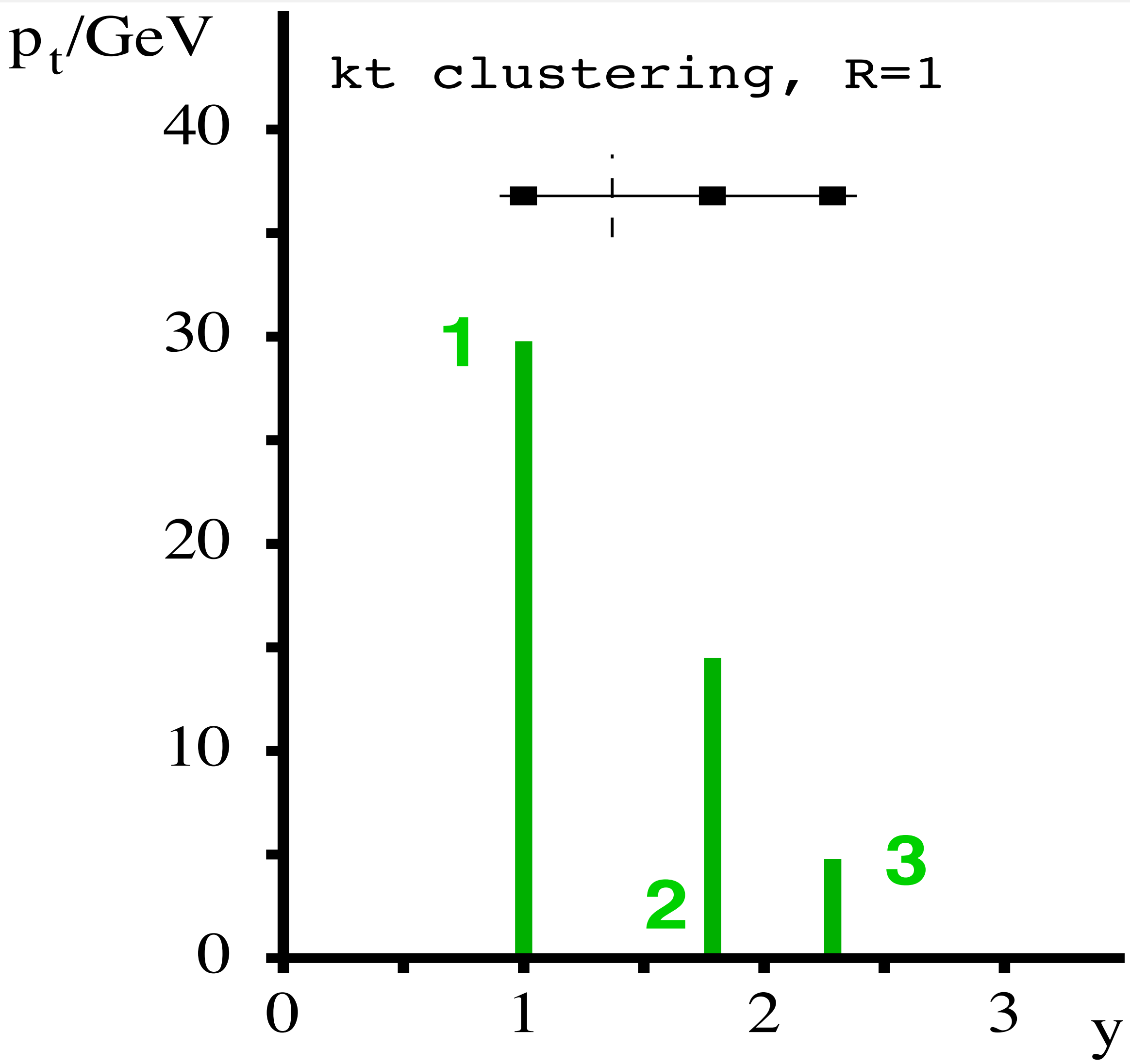


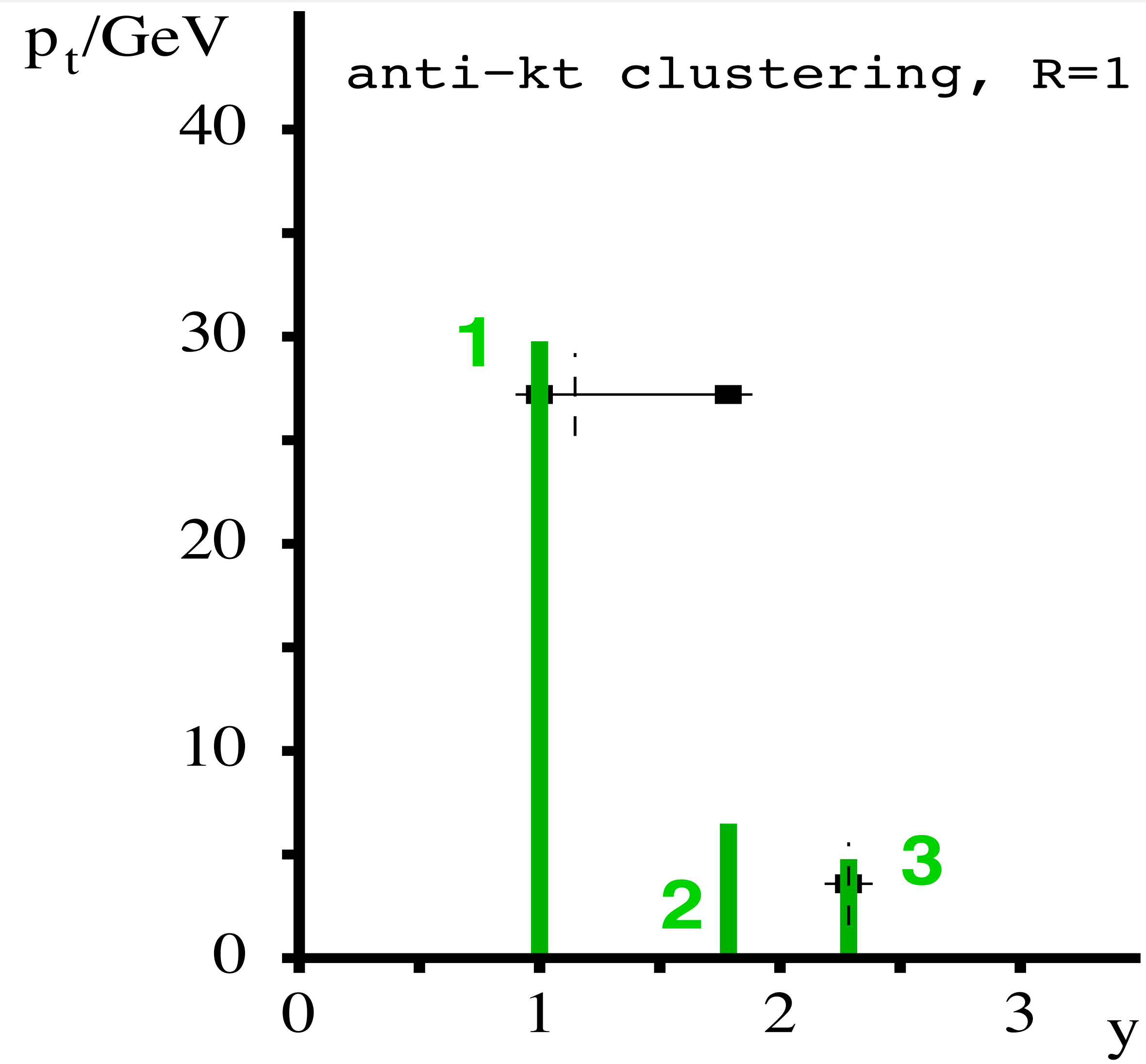
Linearity: k_t v. anti-k_t

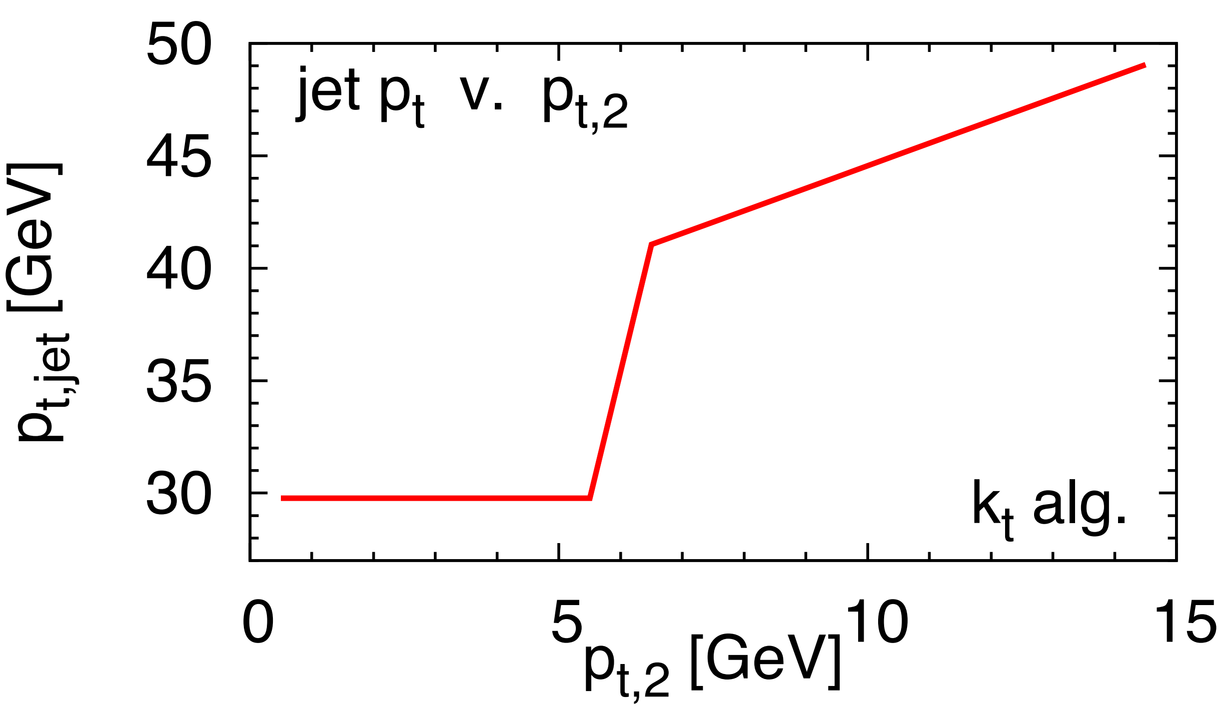


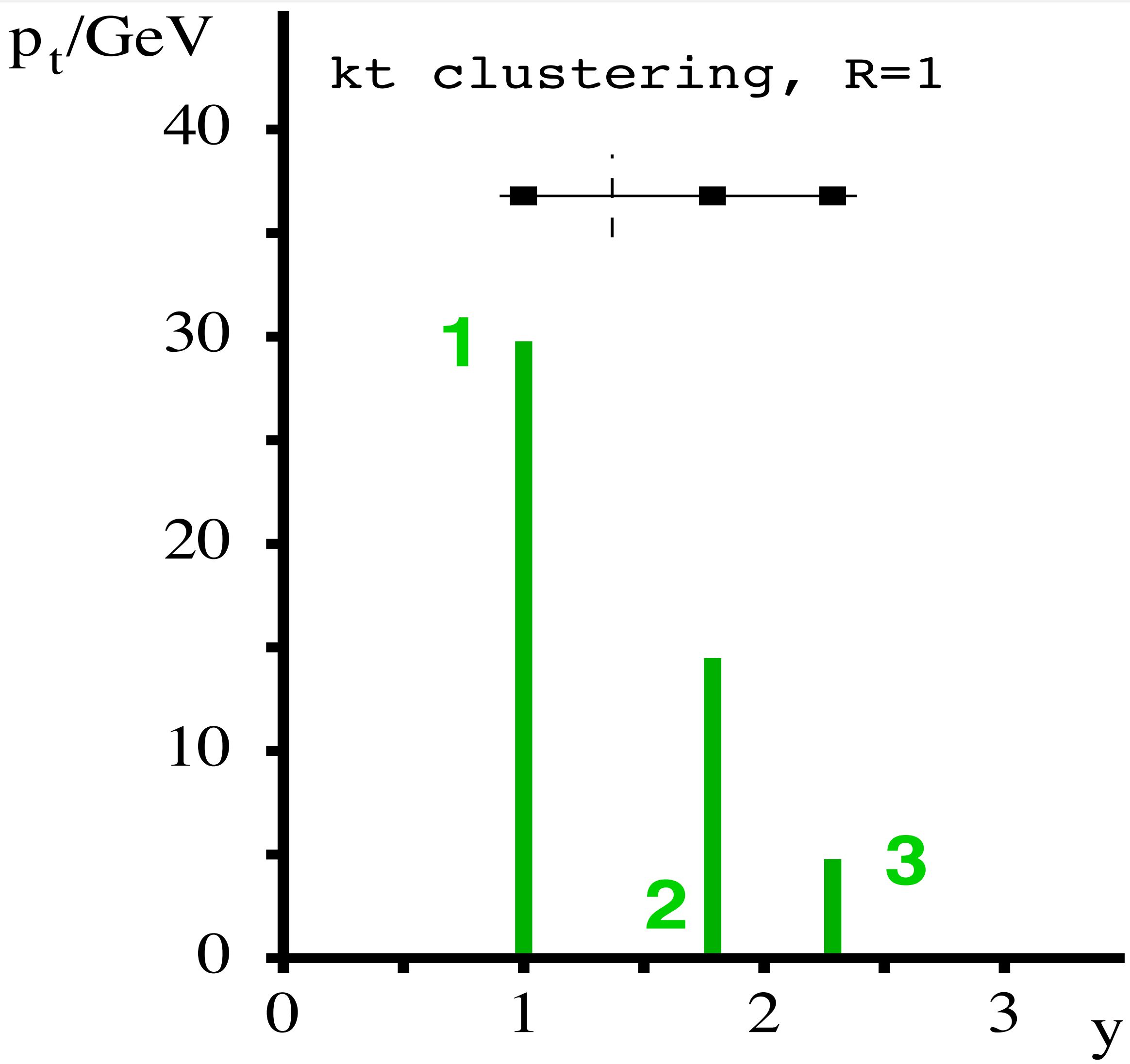


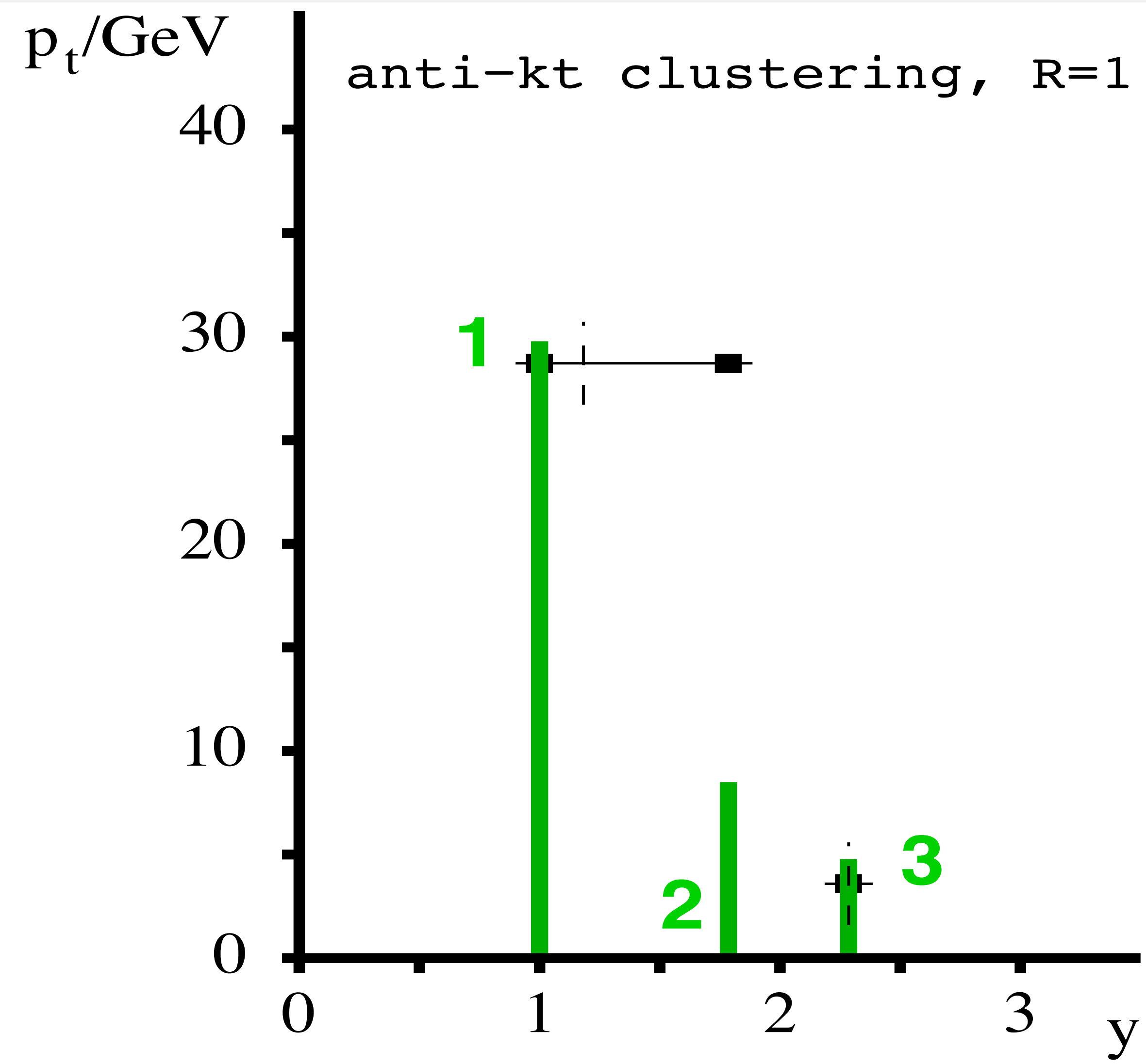


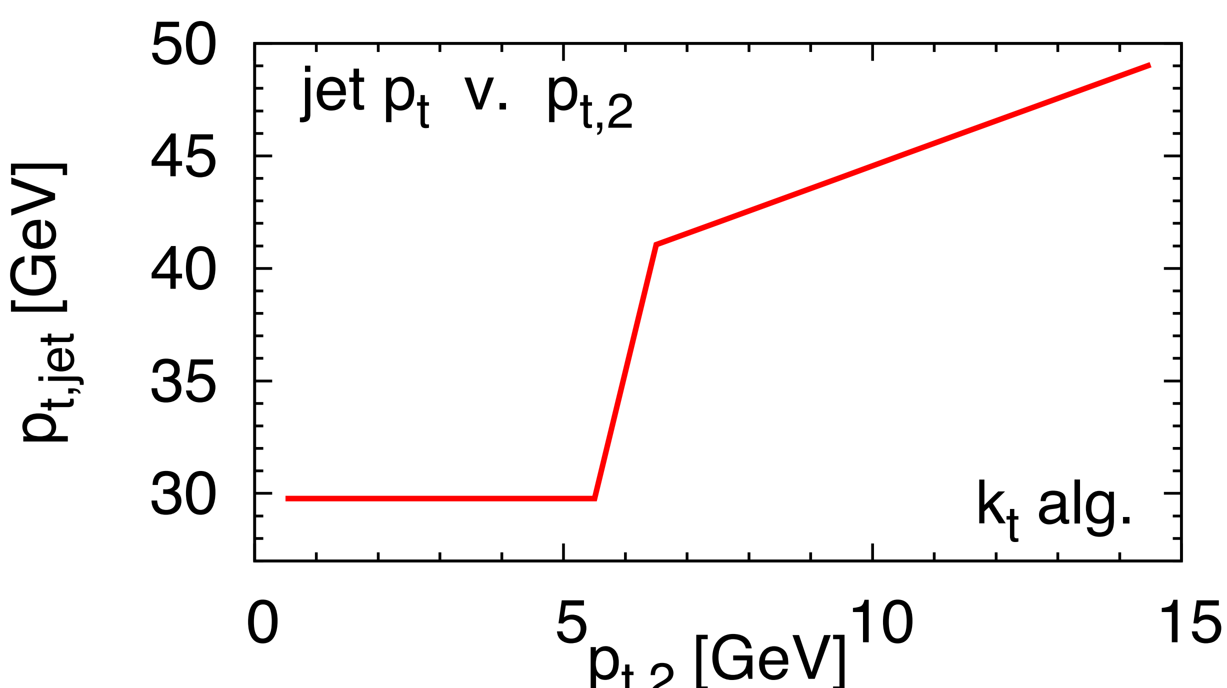

p$_t$/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

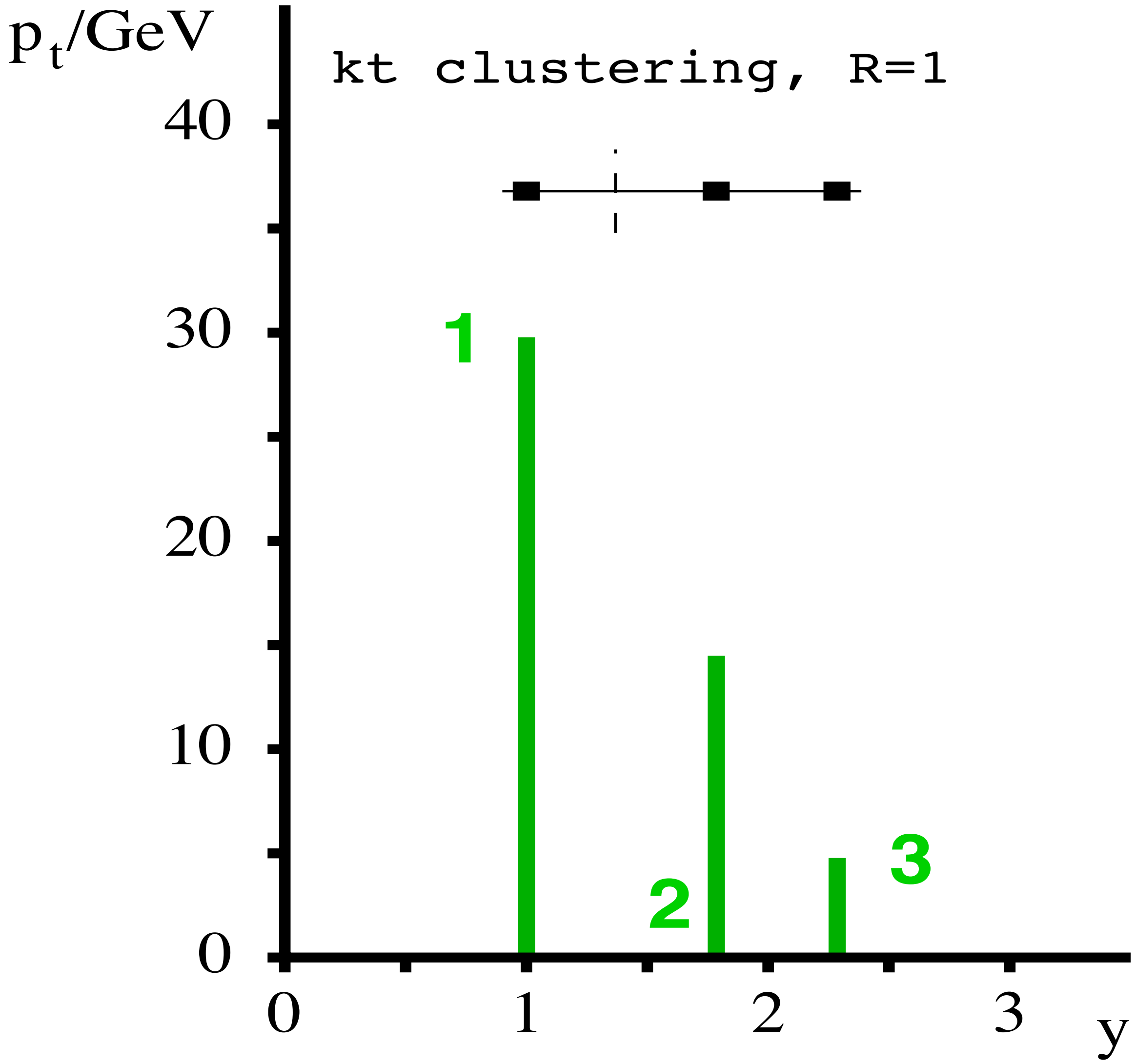

p$_t$/GeV
anti-kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

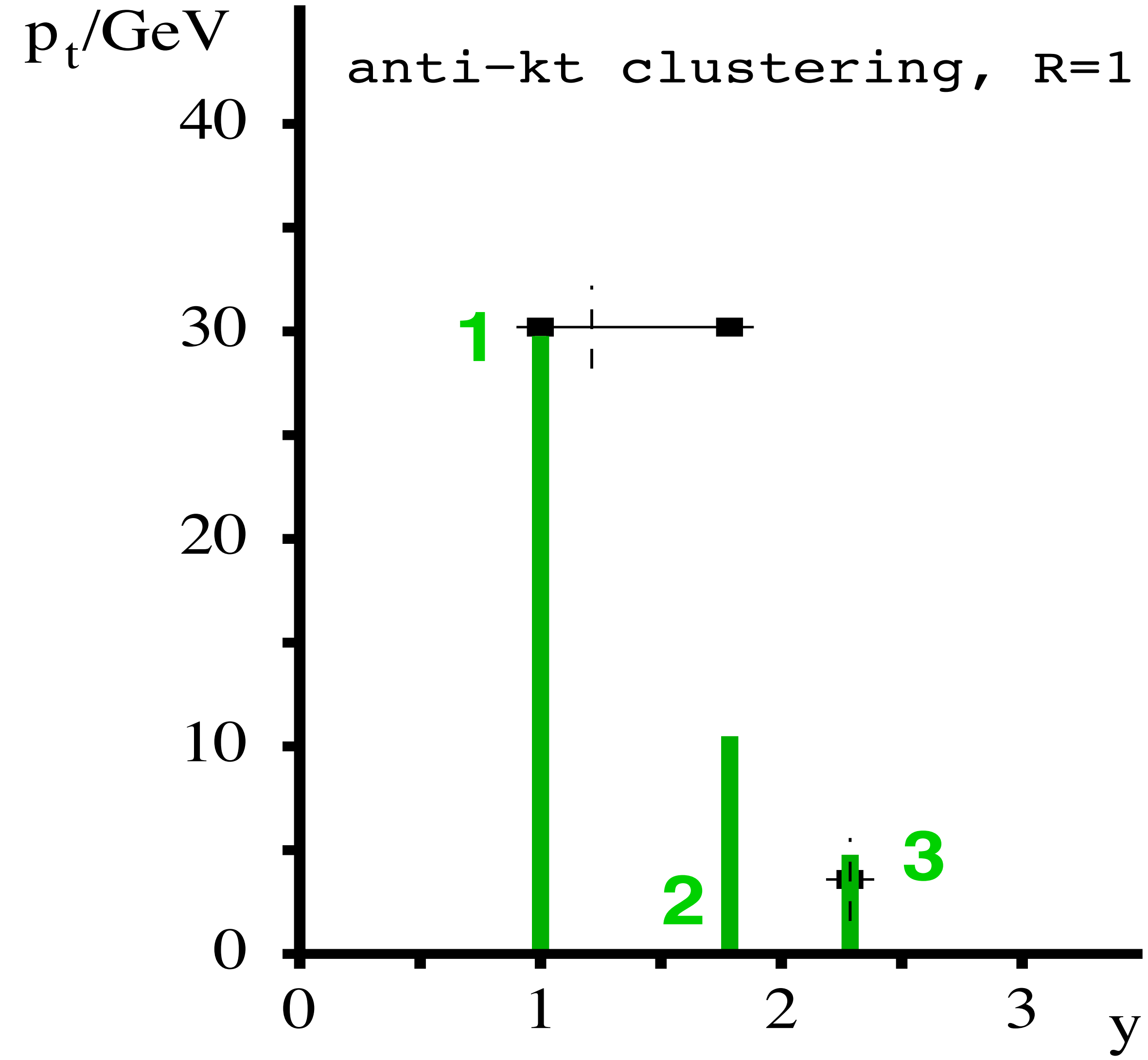

jet p$_t$ v. p$_{t,2}$
k$_t$ alg.
p$_{t,jet}$ [GeV]
50
45
40
35
30
0
5
10
15
p$_{t,2}$ [GeV]

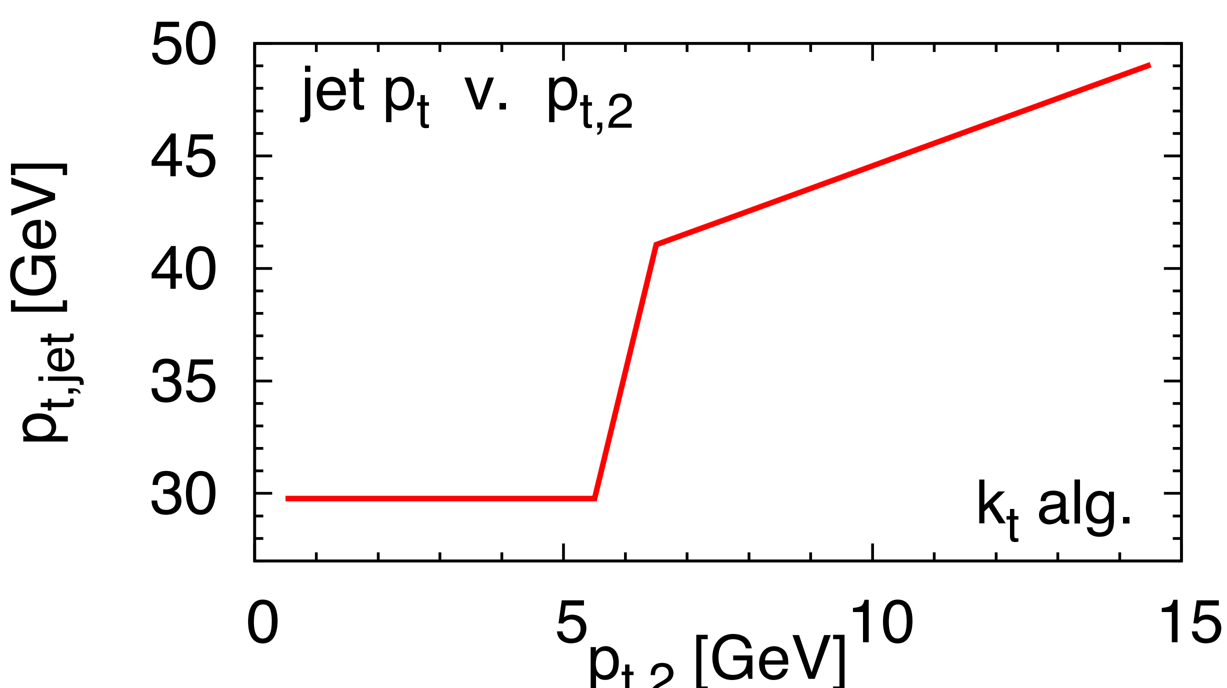

p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

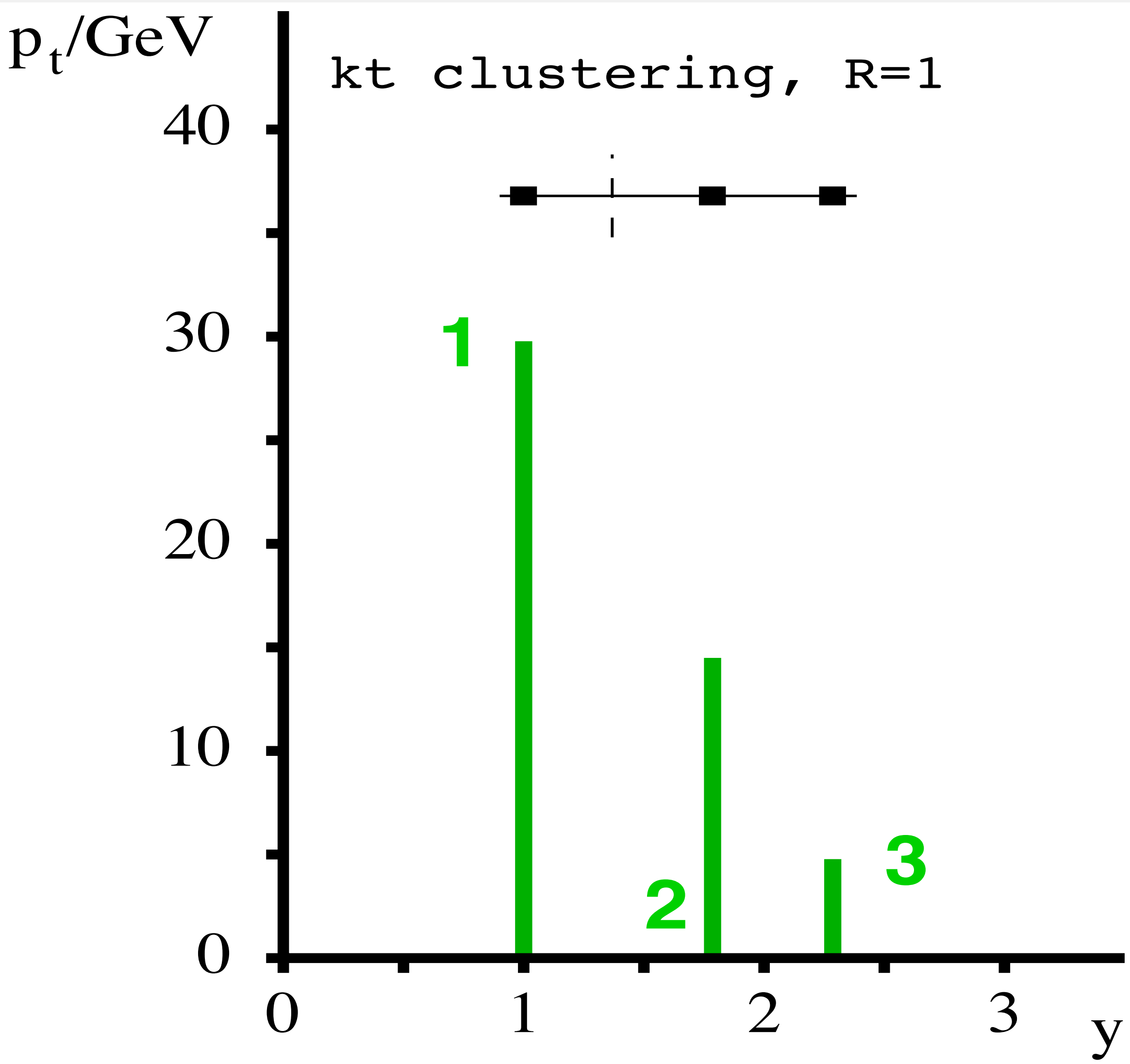

p_t/GeV
anti-kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

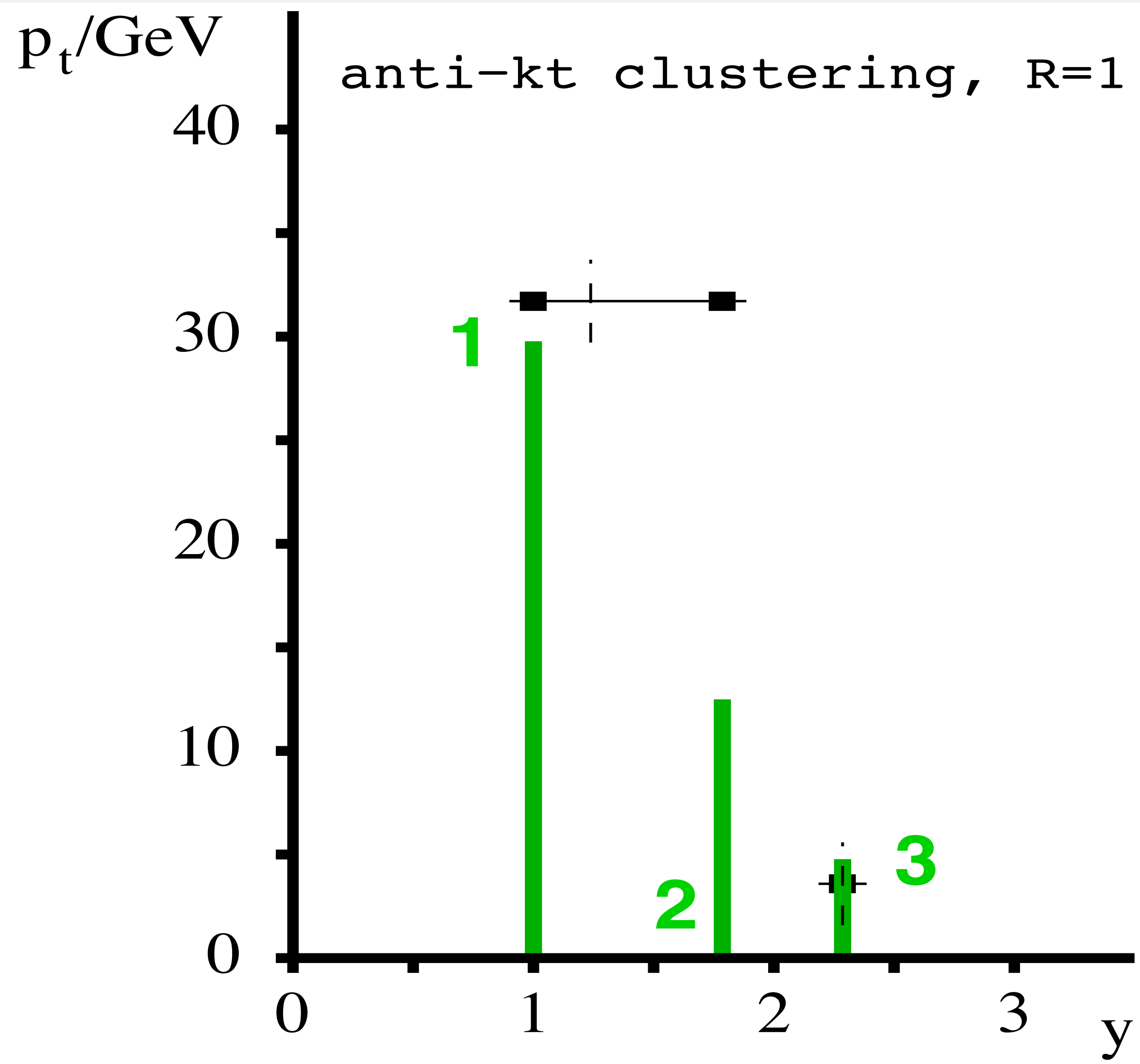

jet p_t v. p_t,2
k_t alg.
p_t,jet [GeV]
p_t,2 [GeV]
50
45
40
35
30
0
5
10
15

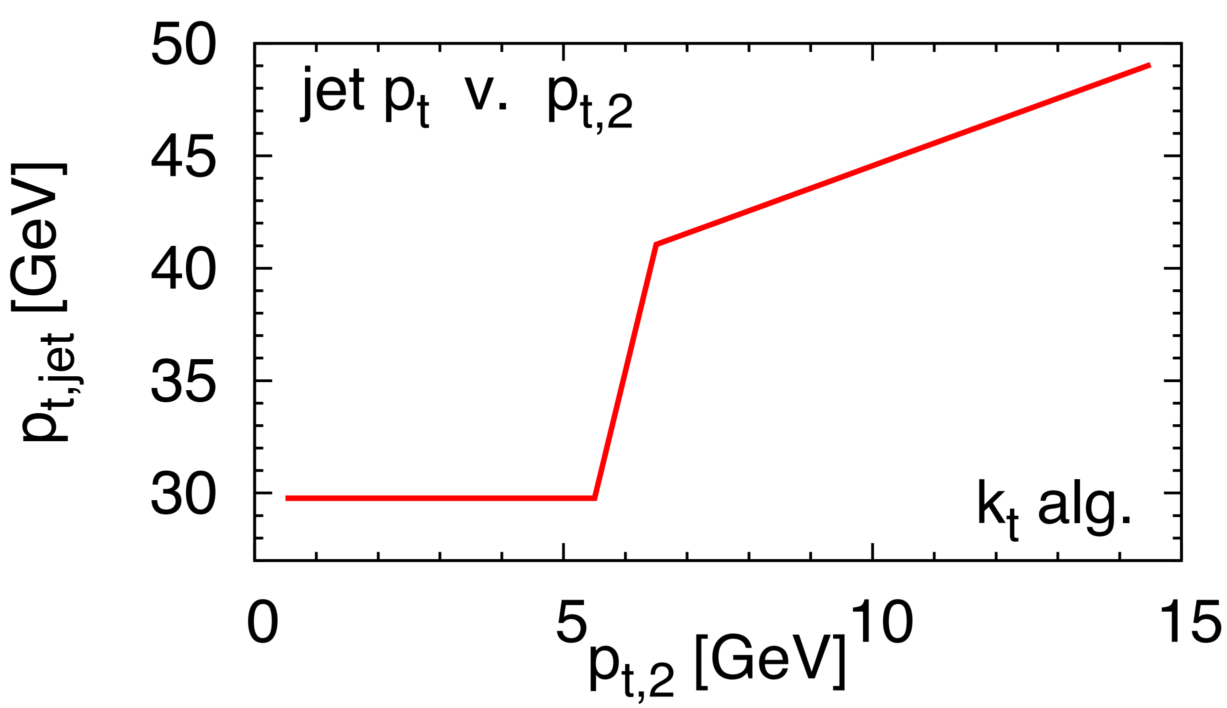

p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

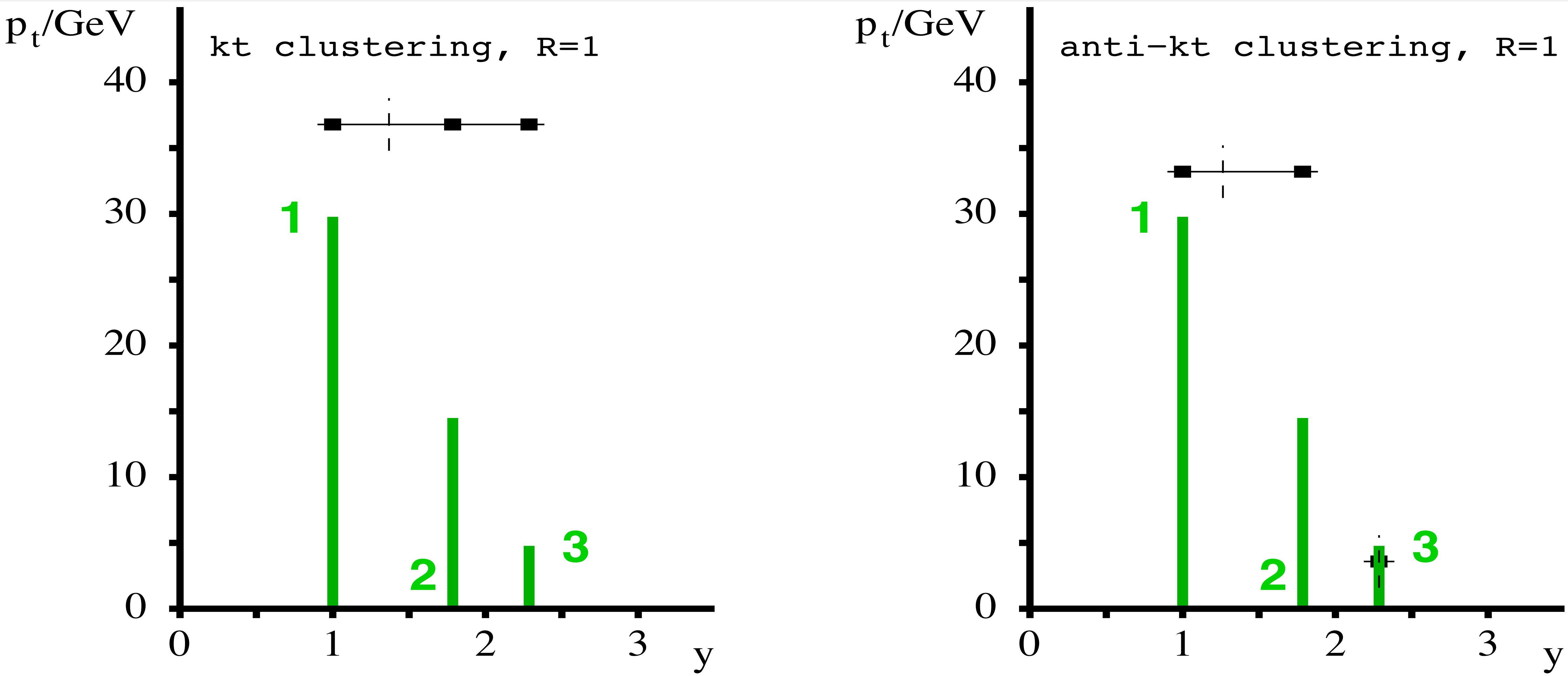

p_t/GeV
anti-kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

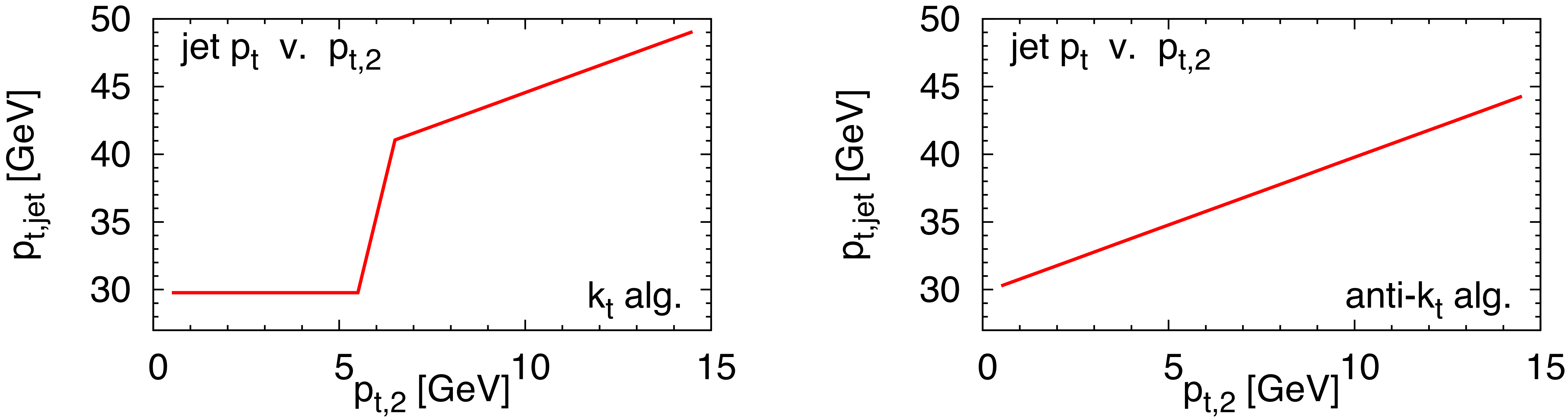

jet p_t v. p_t,2
k_t alg.
p_t,jet [GeV]
p_t,2 [GeV]
50
45
40
35
30
0
5
10
15


p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y

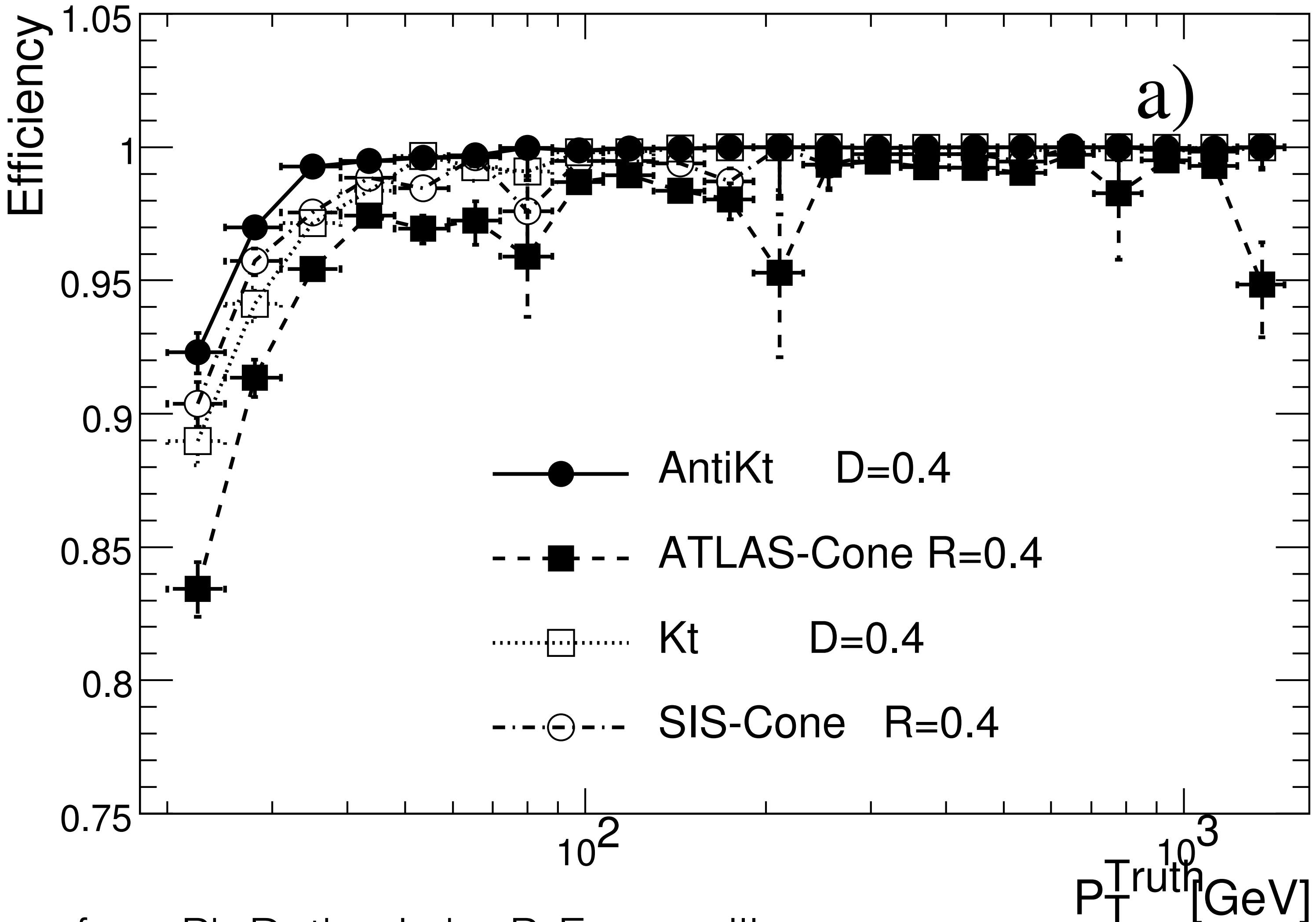

p_t/GeV
anti-kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y


jet p_t v. p_t,2
k_t alg.
p_t,jet [GeV]
50
45
40
35
30
0
5
10
15
p_t,2 [GeV]


p_t/GeV
kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y
p_t/GeV
anti-kt clustering, R=1
40
30
20
10
0
1
2
3
0
1
2
3
y


50
45
40
35
30
jet p_t v. p_t,2
k_t alg.
p_t,jet [GeV]
0
5
10
15
p_t,2 [GeV]
50
45
40
35
30
jet p_t v. p_t,2
anti-k_t alg.
p_t,jet [GeV]
0
5
10
15
p_t,2 [GeV]

Efficiency for finding detector jet that matches particle jet



from Ph.D. thesis by P. Francavilla

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

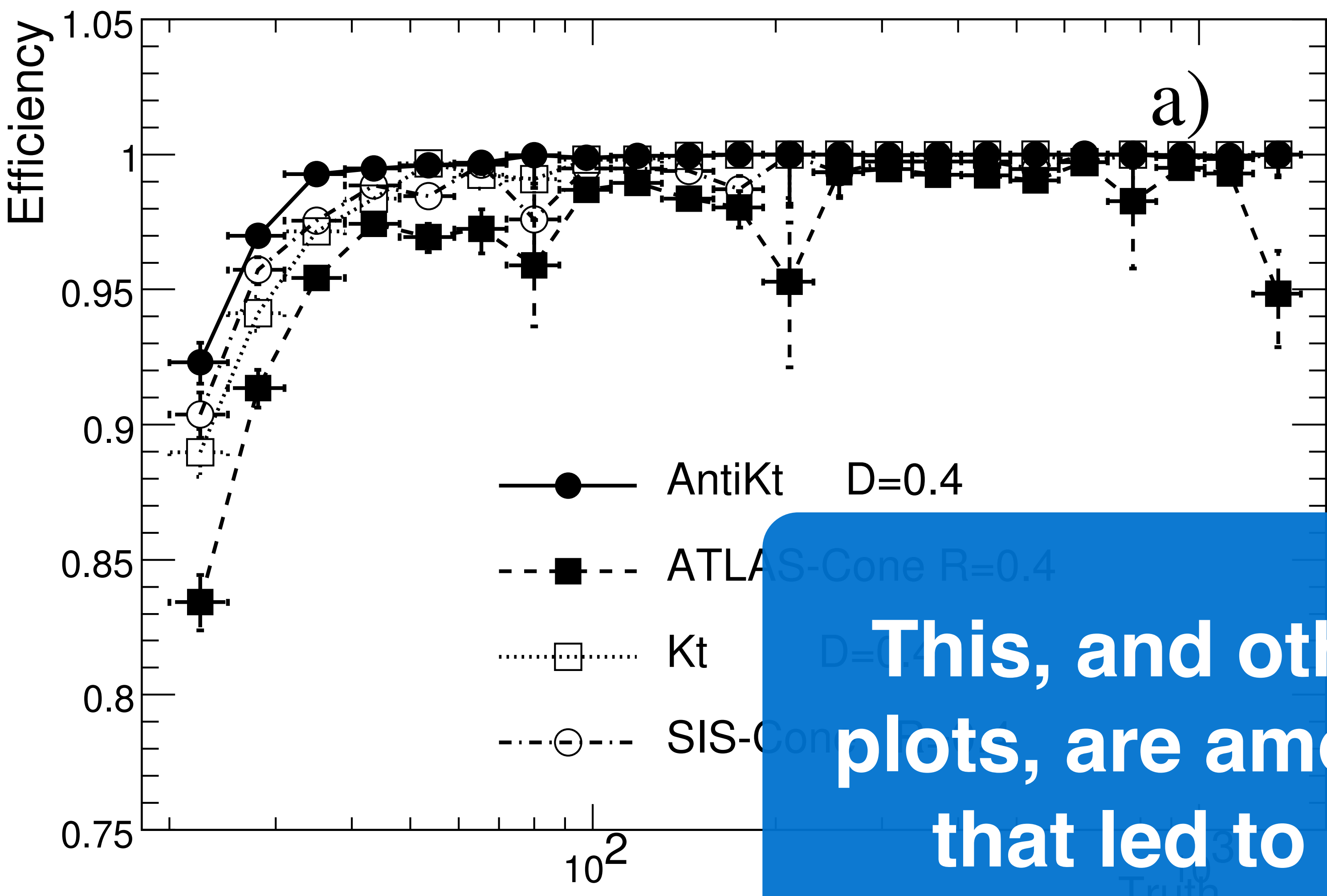
Essentially because anti-k$_t$ has linear response to soft particles.

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

[see blackboard for explanation of plot]

Anti-k_t experimental performance

Efficiency for finding detector jet that matches particle jet



from Ph.D. thesis by P. Francavilla

[see blackboard for explanation of plot]

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

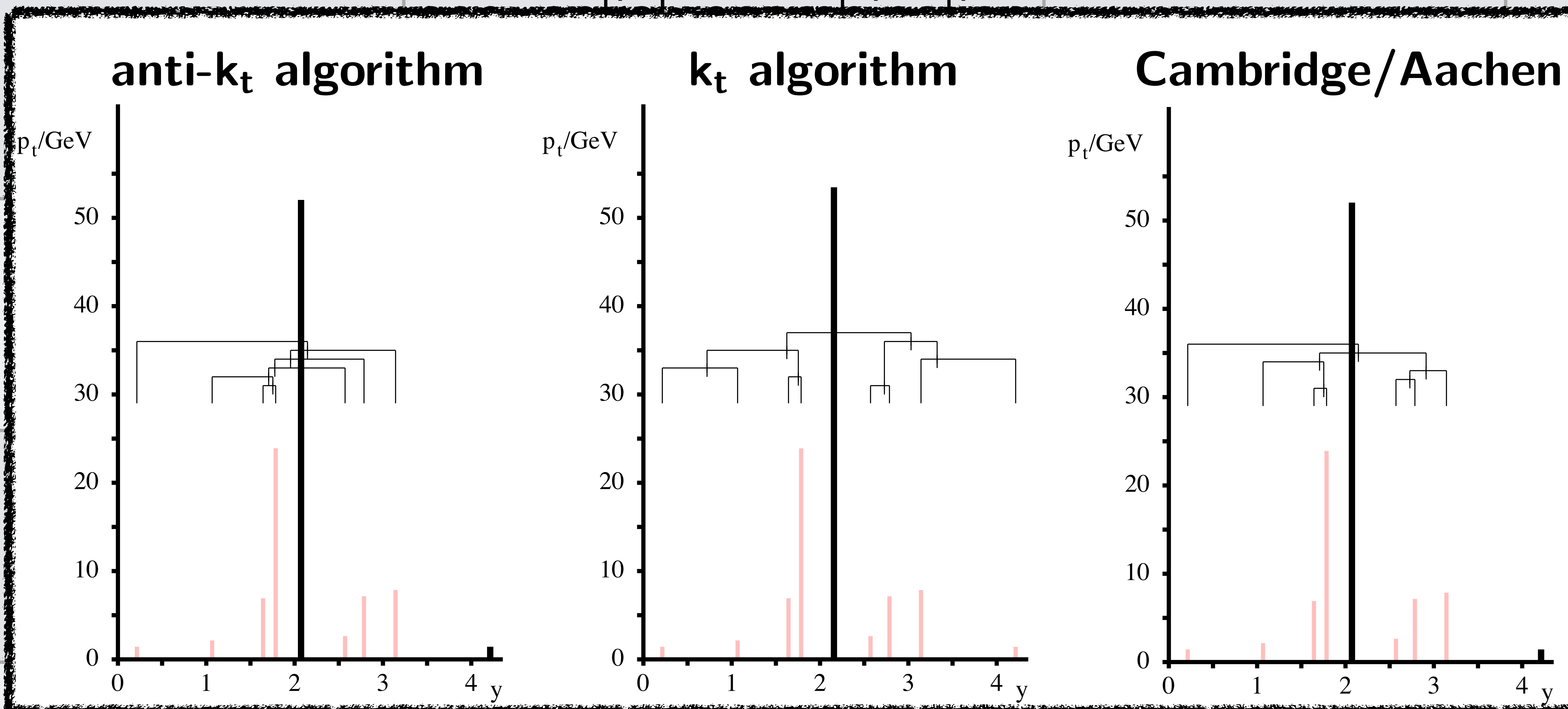
Essentially because anti-k_t has linear response to soft particles.

... ared and ... s needed ... calcs).

This, and other analogous plots, are among the factors that led to anti-k_t being adopted as the main jet algorithm in ATLAS and CMS

Infrared & Collinear safe algorithms

k_t	sequential recombination $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ hierarchical in relative p_t	Catani et al '91 Ellis, Soper '93	NlnN
Cambridge/ Aachen	sequential recombination $d_{ij} = \Delta R_{ij}^2/R^2$ hierarchical in angle	Dokshitzer et al '97 Wengler, Wobish '98	NlnN
anti-k_t	sequential recombination $d_{ij} = \min(k_{ti}^{-2}, k_{tj}^{-2})\Delta R_{ij}^2/R^2$ gives perfectly conical hard jets	MC, Salam, Soyez '08 (Delsart)	$N^{3/2}$
SISCone	Seedless iterative cone with split-merge gives 'economical' jets	Salam, Soyez '07	$N^2\ln N$



SISCone	Seedless iterative cone with split-merge gives 'economical' jets	Salam, Soyez '07	$N^2 \ln N$

Infrared & Collinear safe algorithms

k_t	sequential recombination $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ hierarchical in relative p_t	Catani et al '91 Ellis, Soper '93	NlnN
Cambridge/ Aache	sequential recombination	Dokshitzer et al '97	NlnN
anti-k			$N^{3/2}$
SISCo			$N^2 \ln N$

Advertisment

All are available in FastJet, http://fastjet.fr

(as well as many IRC unsafe ones)

Other (recent) software

SpartyJet (python/root interface to FastJet)

SlowJet (in Pythia8)

SlowJet (mathematica code by J. Ruderman)

```
// specify a jet definition
JetDefinition jet_def(antikt_algorithm, double R);
```

jet_algorithm can be any one of the four IRC safe algorithms, or also most of the old IRC-unsafe ones, for legacy purposes

```
// specify a jet definition
JetDefinition jet_def(antikt_algorithm, double R);
```

jet_algorithm can be any one of the four IRC safe algorithms, or also most of the old IRC-unsafe ones, for legacy purposes

```
// specify the input particles
vector<PseudoJet> input_particles = . . .;
```

```
// specify a jet definition
JetDefinition jet_def(antikt_algorithm, double R);
```

jet_algorithm can be any one of the four IRC safe algorithms, or also most of the old IRC-unsafe ones, for legacy purposes

```
// specify the input particles
vector<PseudoJet> input_particles = . . .;
```

```
// create a ClusterSequence
ClusterSequence cs(input_particles, jet_def);
```

```
// specify a jet definition
JetDefinition jet_def(antikt_algorithm, double R);
```

jet_algorithm can be any one of the four IRC safe algorithms, or also most of the old IRC-unsafe ones, for legacy purposes

```
// specify the input particles
vector<PseudoJet> input_particles = . . .;
```

```
// create a ClusterSequence
ClusterSequence cs(input_particles, jet_def);
```

```
// extract the jets
vector<PseudoJet> jets = sorted_by_pt(cs.inclusive(jets));

// pt of hardest jet
double pt_hardest = jets[0].pt();

// constituents of hardest jet
vector<PseudoJet> constituents = jets[0].constituents();
```

FastJet speed

Time needed to cluster an event with N particles