Jets, our window on partons at the LHC

Gavin Salam

LPTHE, Universities of Paris VI and VII and CNRS

Work (in progress) with M. Cacciari,
M. Dasgupta, L. Magnea, G. Soyez

LPT Orsay
10 janvier 2008

Partons — quarks and gluons — are key concepts of QCD.

- Lagrangian is in terms of quark and gluon fields
- Perturbative QCD *only* deals with partons
- Concept of parton powerful even beyond perturbation theory

hadron classifications

exotic states, e.g. colour glass condensate (high gluon densities)

Yet it is surprisingly hard to give unambiguous meaning to partons.

- Not an asymptotic state of the theory — because of confinemen
- But also even in perturbation theory

because of collinear divergences (in massless approx.)

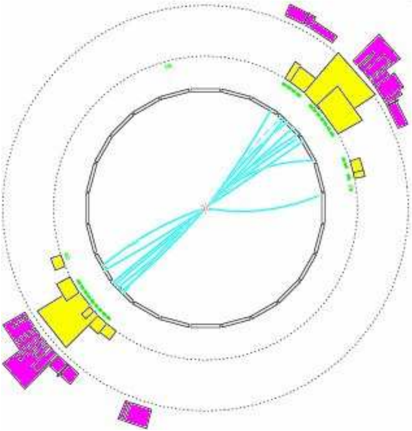
Despite this, there are two decent ways of "seeing" partons:

- Scatter some hard probe off them, e.g. a virtual photon $\longrightarrow$ DIS
- See traces of them in the final state $\longrightarrow$ jets

In each case ill-defined nature of a parton translates into ambiguity in the partonic interpretation of what you see

richness of the physics

Seeing v. defining jets

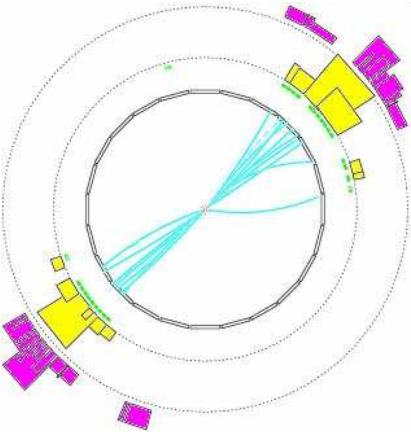


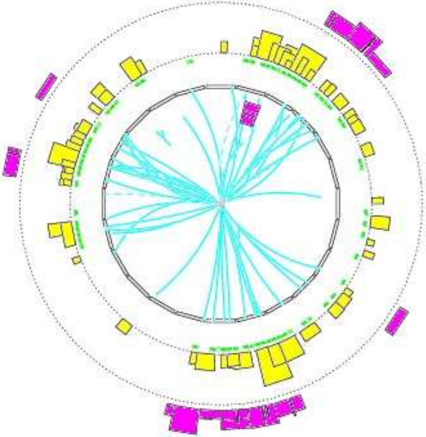
Jets are what we see.
Clearly(?) 2 of them.

2 partons?
$E_{parton} = M_z/2$?

How many jets do you see?
Do you really want to ask yourself this question for 10^8 events?

Seeing v. defining jets





Jets are what we see.
Clearly(?) 2 of them.

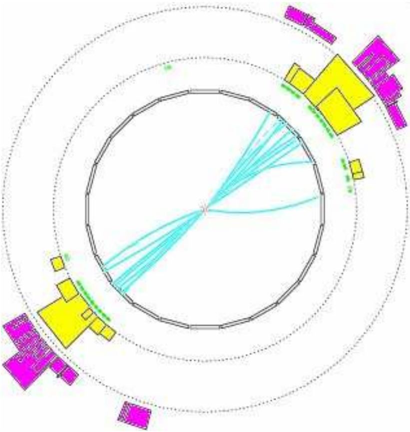
2 partons?

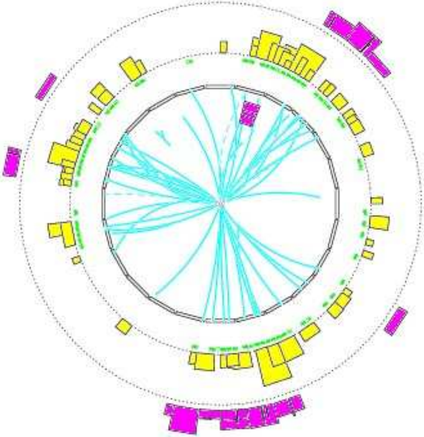
$E_{parton} = M_z/2$?

How many jets do you see?

Do you really want to ask yourself this question for 10^8 events?

Seeing v. defining jets





Jets are what we see.
Clearly(?) 2 of them.

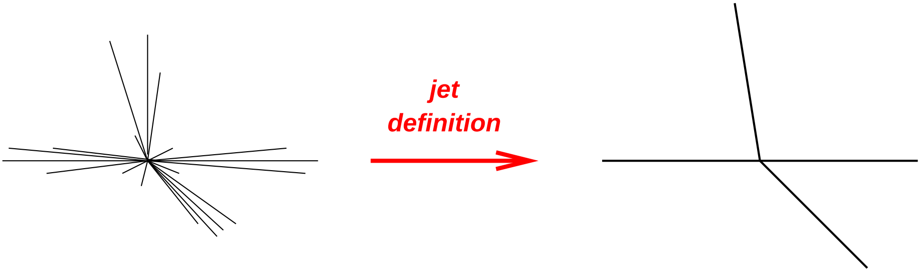
2 partons?

$E_{parton} = M_z/2$?

How many jets do you see?
Do you really want to ask yourself this question for 10^8 events?

Jet definition / algorithm

A jet definition is a systematic procedure that **projects away the multiparticle dynamics**, so as to leave a simple picture of what happened in an event:



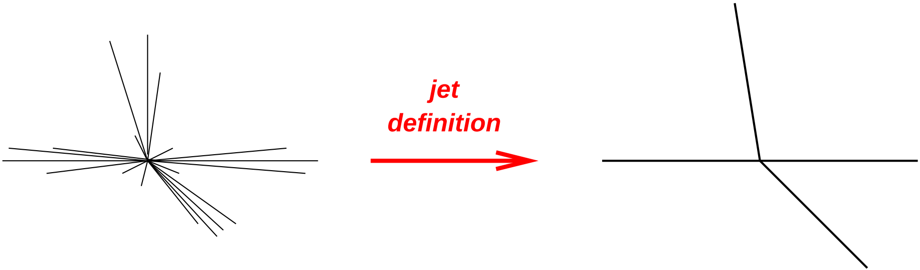
Jets are *as close as we can get to a physical single hard quark or gluon:* with good definitions their properties (multiplicity, energies, [flavour]) are

- finite at any order of perturbation theory
- insensitive to the parton → hadron transition

NB: finiteness ⟷ set of jets depends on jet def.

Jet definition / algorithm

A jet definition is a systematic procedure that **projects away the multiparticle dynamics**, so as to leave a simple picture of what happened in an event:



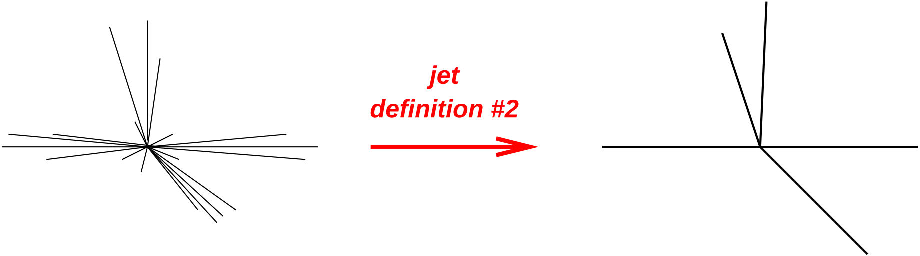
Jets are *as close as we can get to a physical single hard quark or gluon:* with good definitions their properties (multiplicity, energies, [flavour]) are

- finite at any order of perturbation theory
- insensitive to the parton → hadron transition

NB: finiteness ⟷ set of jets depends on jet def.

Jet definition / algorithm

A jet definition is a systematic procedure that **projects away the multiparticle dynamics**, so as to leave a simple picture of what happened in an event:

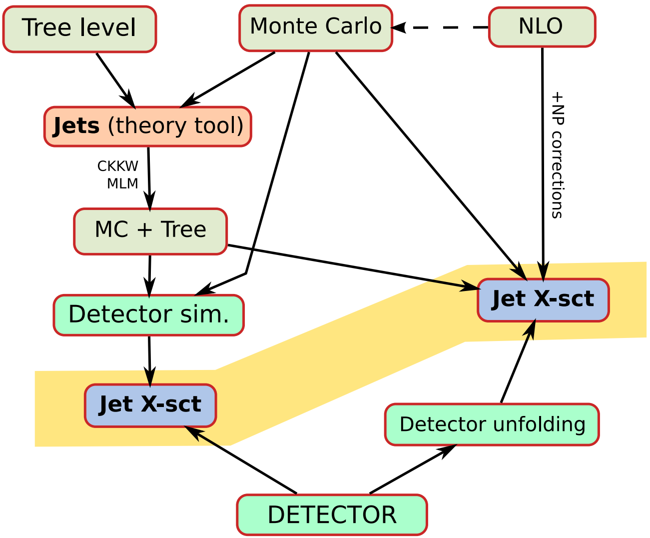


Jets are *as close as we can get to a physical single hard quark or gluon:* with good definitions their properties (multiplicity, energies, [flavour]) are

- finite at any order of perturbation theory
- insensitive to the parton $\rightarrow$ hadron transition

NB: finiteness $\longleftrightarrow$ set of jets depends on jet def.

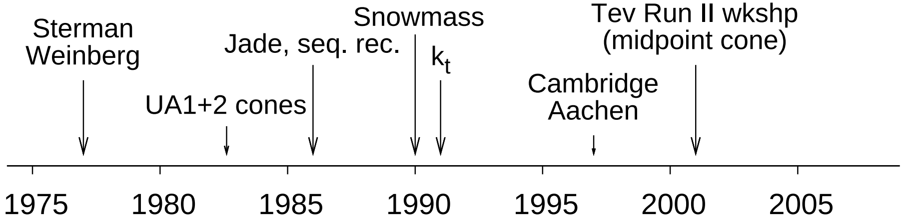
QCD flowchart



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

Jet Definition History

- Periodic key developments in jet definitions spurred by ever-increasing experimental/theoretical sophistication.
- Approach of LHC provides motivation for taking a new, fresh, systematic look at jets.
- This talk: **some of the discoveries along the way**

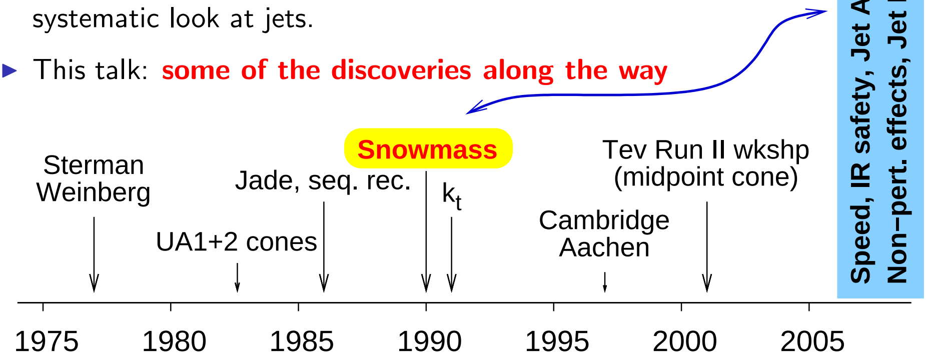


Definitions shown are those with widest exptl. impact

NB: also ARCLUS, OJF, . . .

Jet Definition History

- Periodic key developments in jet definitions spurred by ever-increasing experimental/theoretical sophistication.
- Approach of LHC provides motivation for taking a new, fresh, systematic look at jets.
- This talk: **some of the discoveries along the way**



Definitions shown are those with widest exptl. impact

NB: also ARCLUS, OJF, . . .

What's new for jets @ LHC?

Number of particles:

Experiment	N
LEP, HERA	50
Tevatron	100–400
LHC low-lumi	800
LHC high-lumi	4000
LHC PbPb	30000

- Range & complexity of signatures (jets, $t\bar{t}$, tj, Wj, Hj, $t\bar{t}j$, WWj, Wjj, SUSY, etc.)
- e.g. ~ 5 million $t\bar{t} \to 6$ jet events/year
- Theory investment

~ 100 people $\times$ 10 years
60 – 100 million \$

Physics scales:

Experiment	Physics	Scale
LEP, HERA	Electroweak	100 GeV
	+ Hadronisation	0.5 GeV
Tevatron $\to$ LHC	+ Underlying event	$4 \to 15$ GeV?
LHC	+ BSM	1 TeV?
	+ Pileup	30 – 120 GeV

What's new for jets @ LHC?

Number of particles:

Experiment	N
LEP, HERA	50
Tevatron	100–400
LHC low-lumi	800
LHC high-lumi	4000
LHC PbPb	30000

- Range & complexity of signatures (jets, $t\bar{t}$, tj, Wj, Hj, $t\bar{t}j$, WWj, Wjj, SUSY, etc.)
- e.g. ~ 5 million $t\bar{t} \to 6$ jet events/year
- Theory investment

~ 100 people $\times$ 10 years

$60 - 100$ million \$

Physics scales:

Experiment	Physics	Scale
LEP, HERA	Electroweak	100 GeV
	+ Hadronisation	0.5 GeV
Tevatron $\to$ LHC	+ Underlying event	$4 \to 15$ GeV?
LHC	+ BSM	1 TeV?
	+ Pileup	$30 - 120$ GeV

What's new for jets @ LHC?

Number of particles:

Experiment	N
LEP, HERA	50
Tevatron	100–400
LHC low-lumi	800
LHC high-lumi	4000
LHC PbPb	30000

- Range & complexity of signatures (jets, $t\bar{t}$, tj, Wj, Hj, $t\bar{t}j$, WWj, Wjj, SUSY, etc.)
- e.g. ~ 5 million $t\bar{t} \to 6$ jet events/year
- Theory investment

~ 100 people $\times$ 10 years

$60 - 100$ million \$

Physics scales:

Experiment	Physics	Scale
LEP, HERA	Electroweak	100 GeV
	+ Hadronisation	0.5 GeV
Tevatron $\to$ LHC	+ Underlying event	$4 \to 15$ GeV?
LHC	+ BSM	1 TeV?
	+ Pileup	$30 - 120$ GeV

Old issues? 1990 "standards"

Snowmass Accord (1990):

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

Toward a Standardization of Jet Definitions *

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

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

Without these, either the experiment won't use the jet-definition, or the theoretical calculations will be compromised

Long satisfied in e^+e^- and DIS

Satisfied in $\lesssim 5\%$ of jet work at Tevatron

Hardly discussed in LHC TDRs

Old issues? 1990 “standards”

Snowmass Accord (1990):

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

Toward a Standardization of Jet Definitions *

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

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

Without these, either the experiment won’t use the jet-definition, or the theoretical calculations will be compromised

Long satisfied in e^+e^- and DIS

Satisfied in $\lesssim 5\%$ of jet work at Tevatron

Hardly discussed in LHC TDRs

Two classes of jet algorithm

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

Both had serious issues that got in way of practical use and/or physical validity

Two classes of jet algorithm

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

Both had serious issues that got in way of practical use and/or physical validity

Sequential recombination algorithms

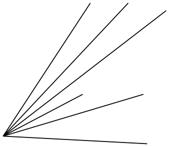
k_t algorithm — Catani, Dokshizter, Olsson, Seymour, Turnock, Webber '91–'93
Ellis, Soper '93

- Find smallest of all $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ and $d_{iB} = k_i^2$
- Recombine i, j (if iB: $i \to$ jet)
- Repeat

NB: hadron collider variables

- $\Delta R_{ij}^2 = (\phi_i - \phi_j)^2 + (y_i - y_j)^2$
- rapidity $y_i = \frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$
- ΔR_{ij} is boost invariant angle

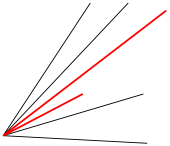
R sets jet opening angle



Sequential recombination algorithms

k_t algorithm — Catani, Dokshizter, Olsson, Seymour, Turnock, Webber '91–'93
Ellis, Soper '93

- Find smallest of all $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ and $d_{iB} = k_i^2$
- Recombine i, j (if iB: $i \to$ jet)
- Repeat



NB: hadron collider variables

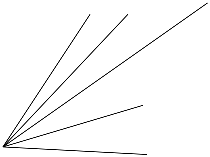
- $\Delta R_{ij}^2 = (\phi_i - \phi_j)^2 + (y_i - y_j)^2$
- rapidity $y_i = \frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$
- ΔR_{ij} is boost invariant angle

R sets jet opening angle

Sequential recombination algorithms

k_t algorithm Catani, Dokshizter, Olsson, Seymour, Turnock, Webber '91–'93
Ellis, Soper '93

- Find smallest of all $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ and $d_{iB} = k_i^2$
- Recombine i, j (if iB: $i \to$ jet)
- Repeat



NB: hadron collider variables

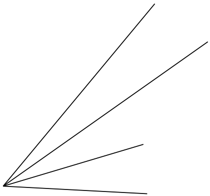
- $\Delta R_{ij}^2 = (\phi_i - \phi_j)^2 + (y_i - y_j)^2$
- rapidity $y_i = \frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$
- ΔR_{ij} is boost invariant angle

R sets jet opening angle

Sequential recombination algorithms

k_t algorithm — Catani, Dokshizter, Olsson, Seymour, Turnock, Webber '91–'93
Ellis, Soper '93

- Find smallest of all $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ and $d_{iB} = k_i^2$
- Recombine i, j (if iB: $i \to$ jet)
- Repeat



NB: hadron collider variables

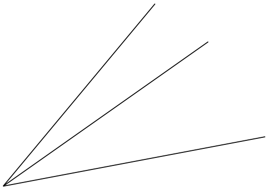
- $\Delta R_{ij}^2 = (\phi_i - \phi_j)^2 + (y_i - y_j)^2$
- rapidity $y_i = \frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$
- ΔR_{ij} is boost invariant angle

R sets jet opening angle

Sequential recombination algorithms

k_t algorithm — Catani, Dokshizter, Olsson, Seymour, Turnock, Webber '91–'93; Ellis, Soper '93

- Find smallest of all $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ and $d_{iB} = k_i^2$
- Recombine i, j (if iB: $i \to$ jet)
- Repeat



NB: hadron collider variables

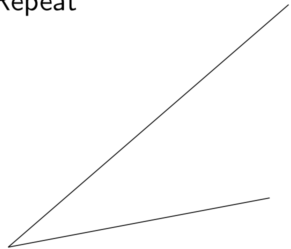
- $\Delta R_{ij}^2 = (\phi_i - \phi_j)^2 + (y_i - y_j)^2$
- rapidity $y_i = \frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$
- ΔR_{ij} is boost invariant angle

R sets jet opening angle

Sequential recombination algorithms

k_t algorithm — Catani, Dokshizter, Olsson, Seymour, Turnock, Webber '91–'93
Ellis, Soper '93

- Find smallest of all $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ and $d_{iB} = k_i^2$
- Recombine i, j (if iB: $i \rightarrow$ jet)
- Repeat



NB: hadron collider variables

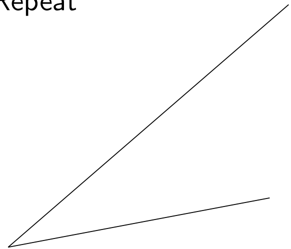
- $\Delta R_{ij}^2 = (\phi_i - \phi_j)^2 + (y_i - y_j)^2$
- rapidity $y_i = \frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$
- ΔR_{ij} is boost invariant angle

R sets jet opening angle

Sequential recombination algorithms

k_t algorithm — Catani, Dokshizter, Olsson, Seymour, Turnock, Webber '91–'93; Ellis, Soper '93

- Find smallest of all $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ and $d_{iB} = k_i^2$
- Recombine i, j (if iB: $i \to$ jet)
- Repeat



NB: hadron collider variables

- $\Delta R_{ij}^2 = (\phi_i - \phi_j)^2 + (y_i - y_j)^2$
- rapidity $y_i = \frac{1}{2} \ln \frac{E_i + p_{zi}}{E_i - p_{zi}}$
- ΔR_{ij} is boost invariant angle

R sets jet opening angle

Why k_t?

k_t distance measures

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

are closely related to structure of divergences for QCD emissions

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \sim \frac{\alpha_s C_A}{2\pi}\frac{dk_{tj}}{\min(k_{ti}, k_{tj})}\frac{d\Delta R_{ij}}{\Delta R_{ij}}\,, \qquad (k_{tj} \ll k_{ti}\,,\ \ \Delta R_{ij} \ll 1)$$

and

$$[dk_i]|M^2_{Beam\to Beam+g_i}(k_i)| \sim \frac{\alpha_s C_A}{\pi}\frac{dk_{ti}}{k_{ti}}\,d\eta_i\,, \qquad (k_{ti}^2 \ll \{\hat{s}, \hat{t}, \hat{u}\})$$

k_t algorithm attempts approximate inversion of branching process

Why k_t?

k_t distance measures

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

are closely related to structure of divergences for QCD emissions

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \sim \frac{\alpha_s C_A}{2\pi} \frac{dk_{tj}}{\min(k_{ti}, k_{tj})} \frac{d\Delta R_{ij}}{\Delta R_{ij}}\,, \qquad (k_{tj} \ll k_{ti}\,,\ \ \Delta R_{ij} \ll 1)$$

and

$$[dk_i]|M^2_{Beam\to Beam+g_i}(k_i)| \sim \frac{\alpha_s C_A}{\pi} \frac{dk_{ti}}{k_{ti}}\, d\eta_i\,, \qquad (k_{ti}^2 \ll \{\hat{s}, \hat{t}, \hat{u}\})$$

k_t algorithm attempts approximate inversion of branching process

Computing...

'Trivial' computational issue:

- for N particles: N^2 d_{ij} searched through N times $= N^3$
- 4000 particles (or calo cells): 1 minute

  NB: often study $10^7 - 10^8$ events (20-200 CPU years)
- Heavy Ions: 30000 particles: 10 hours/event

As far as possible physics choices should not be limited by computing.

Even if we're clever about repeating the full search each time, we still have $\mathcal{O}\left(N^2\right)$ d_{ij}'s to establish

k_t is a form of Hierarchical Clustering

Fast Hierarchical Clustering and Other Applications of Dynamic Closest Pairs

David Eppstein
UC Irvine

We develop data structures for dynamic closest pair problems with arbitrary distance functions, that do not necessarily come from any geometric structure on the objects. Based on a technique previously used by the author for Euclidean closest pairs, we show how to insert and delete objects from an n-object set, maintaining the closest pair, in $O(n \log^2 n)$ time per update and $O(n)$ space. With quadratic space, we can instead use a quadtree-like structure to achieve an optimal time bound, $O(n)$ per update. We apply these data structures to hierarchical clustering, greedy matching, and TSP heuristics, and discuss other potential applications in machine learning, Gröbner bases, and local improvement algorithms for partition and placement problems. Experiments show our new methods to be faster in practice than previously used heuristics.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms**]: Nonnumeric Algorithms

General Terms: Closest Pair, Agglomerative Clustering

Additional Key Words and Phrases: TSP, matching, conga line data structure, quadtree, nearest neighbor heuristic

1. INTRODUCTION

Hierarchical clustering has long been a mainstay of statistical analysis, and clustering based methods have attracted attention in other fields: computational biology (reconstruction of evolutionary trees; tree-based multiple sequence alignment), scientific simulation (n-body problems), theoretical computer science (network design and nearest neighbor searching) and of course the web (hierarchical indices such as Yahoo). Many clustering methods have been devised and used in these applications, but less effort has gone into algorithmic speedups of these methods.

In this paper we identify and demonstrate speedups for a key subroutine used in several clustering algorithms, that of maintaining closest pairs in a dynamic set of objects. We also describe several other applications or potential applications of the

k_t alg. is so good it's used throughout science!

NB HEP is not only field to use brute-force. . .

For general distance measures problem reduces to $\sim N^2$ (factor $\sim$ 20 for $N = 1000$).

Eppstein '99
+ Cardinal '03

k_t is a form of Hierarchical Clustering

Fast Hierarchical Clustering and Other Applications of Dynamic Closest Pairs

David Eppstein
UC Irvine

We develop data structures for dynamic closest pair problems with arbitrary distance functions, that do not necessarily come from any geometric structure on the objects. Based on a technique previously used by the author for Euclidean closest pairs, we show how to insert and delete objects from an n-object set, maintaining the closest pair, in $O(n \log^2 n)$ time per update and $O(n)$ space. With quadratic space, we can instead use a quadtree-like structure to achieve an optimal time bound, $O(n)$ per update. We apply these data structures to hierarchical clustering, greedy matching, and TSP heuristics, and discuss other potential applications in machine learning, Gröbner bases, and local improvement algorithms for partition and placement problems. Experiments show our new methods to be faster in practice than previously used heuristics.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms**]: Nonnumeric Algorithms

General Terms: Closest Pair, Agglomerative Clustering

Additional Key Words and Phrases: TSP, matching, conga line data structure, quadtree, nearest neighbor heuristic

Of these naive methods, brute force recomputation may be most commonly used, due to its low space requirements and ease of implementation. Three hierarchical clustering codes we examined, Zupan's [Zupan 1982], CLUSTAL W [Thompson et al. 1994], and PHYLIP [Felsenstein 1995] use brute force. (Indeed, they do not even save space by doing so, since they all store the distance matrix.) Pazzani's learning code [Pazzani 1997] also uses brute force (M. Pazzani, personal communication), as does *Mathematica*'s Gröbner basis code (D. Lichtblau, personal communication).

k_t alg. is so good it's used throughout science!

NB HEP is not only field to use brute-force. . .

For general distance measures problem reduces to $\sim N^2$ (factor ~ 20 for $N = 1000$).

Eppstein '99
+ Cardinal '03

Can we do better than N^2?

There are $N(N-1)/2$ distances d_{ij} — surely we have to calculate them all in order to find smallest?

k_t distance measure is partly *geometrical:*

- Consider smallest $d_{ij} = \min(k_{ti}^2, k_{tj}^2) R_{ij}^2$
- Suppose $k_{ti} < k_{tj}$
- Then: $R_{ij} \leq R_{i\ell}$ for any $\ell \neq j$. [If $\exists\ \ell$ s.t. $R_{i\ell} < R_{ij}$ then $d_{i\ell} < d_{ij}$]

In words: if i, j form smallest d_{ij} then j is geometrical nearest neighbour (GNN) of i.

k_t distance need only be calculated between GNNs

Each point has 1 GNN → need only calculate N d_{ij}'s

Can we do better than N^2?

There are $N(N-1)/2$ distances d_{ij} — surely we have to calculate them all in order to find smallest?

k_t distance measure is partly *geometrical:*

- Consider smallest $d_{ij} = \min(k_{ti}^2, k_{tj}^2) R_{ij}^2$
- Suppose $k_{ti} < k_{tj}$
- Then: $R_{ij} \leq R_{i\ell}$ for any $\ell \neq j$. [If $\exists\ \ell$ s.t. $R_{i\ell} < R_{ij}$ then $d_{i\ell} < d_{ij}$]

In words: if i, j form smallest d_{ij} then j is geometrical nearest neighbour (GNN) of i.

k_t distance need only be calculated between GNNs

Each point has 1 GNN $\rightarrow$ need only calculate N d_{ij}'s

Can we do better than N^2?

There are $N(N-1)/2$ distances d_{ij} — surely we have to calculate them all in order to find smallest?

k_t distance measure is partly *geometrical:*

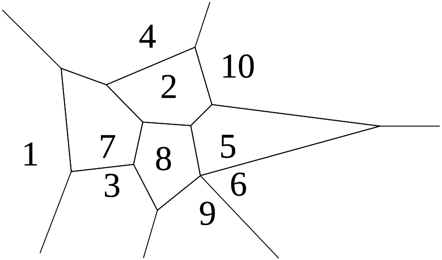
- Consider smallest $d_{ij} = \min(k_{ti}^2, k_{tj}^2) R_{ij}^2$
- Suppose $k_{ti} < k_{tj}$
- Then: $R_{ij} \leq R_{i\ell}$ for any $\ell \neq j$. [If $\exists\ \ell$ s.t. $R_{i\ell} < R_{ij}$ then $d_{i\ell} < d_{ij}$]

In words: if i, j form smallest d_{ij} then j is geometrical nearest neighbour (GNN) of i.

k_t distance need only be calculated between GNNs

Each point has 1 GNN → need only calculate N d_{ij}'s

Finding Geom Nearest Neighbours



Given a set of vertices on plane (1...10) a *Voronoi diagram* partitions plane into cells containing all points closest to each vertex

Dirichlet '1850, Voronoi '1908

A vertex's nearest other vertex is always in an adjacent cell.

E.g. GNN of point 7 will be found among 1,4,2,8,3 (it turns out to be 3)

Construction of Voronoi diagram for N points: $N \ln N$ time — Fortune '88

Update of 1 point in Voronoi diagram: $\ln N$ time

Devillers '99 [+ related work by other authors]

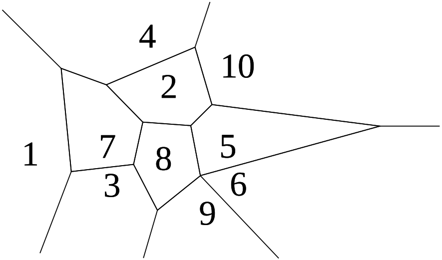
Convenient C++ package available: **CGAL** — `http://www.cgal.org`

Assemble with other comp. science methods: **FastJet**

Cacciari & GPS, hep-ph/0512210

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

Finding Geom Nearest Neighbours



Given a set of vertices on plane (1...10) a *Voronoi diagram* partitions plane into cells containing all points closest to each vertex

Dirichlet '1850, Voronoi '1908

A vertex's nearest other vertex is always in an adjacent cell.

E.g. GNN of point 7 will be found among 1,4,2,8,3 (it turns out to be 3)

Construction of Voronoi diagram for N points: $N \ln N$ time — Fortune '88

Update of 1 point in Voronoi diagram: $\ln N$ time

Devillers '99 [+ related work by other authors]

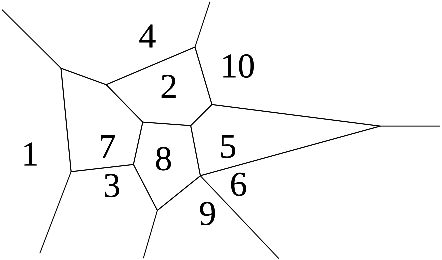
Convenient C++ package available: **CGAL** — http://www.cgal.org

Assemble with other comp. science methods: **FastJet**

Cacciari & GPS, hep-ph/0512210

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

Finding Geom Nearest Neighbours



Given a set of vertices on plane (1...10) a *Voronoi diagram* partitions plane into cells containing all points closest to each vertex

Dirichlet '1850, Voronoi '1908

A vertex's nearest other vertex is always in an adjacent cell.

E.g. GNN of point 7 will be found among 1,4,2,8,3 (it turns out to be 3)

Construction of Voronoi diagram for N points: $N \ln N$ time — Fortune '88

Update of 1 point in Voronoi diagram: $\ln N$ time

Devillers '99 [+ related work by other authors]

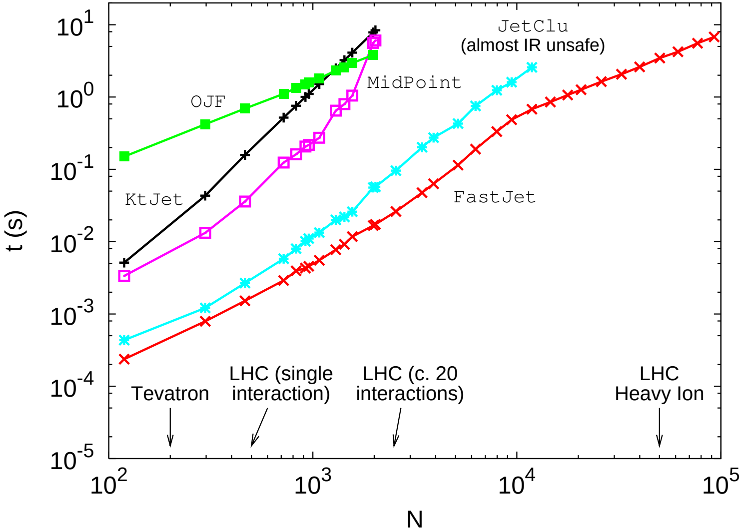
Convenient C++ package available: **CGAL** — `http://www.cgal.org`

Assemble with other comp. science methods: **FastJet**

Cacciari & GPS, hep-ph/0512210

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

FastJet performance



NB: for $N < 10^4$, FastJet switches to a related geometrical N^2 alg.

Conclusion: speed issues for k_t resolved

Cone basics

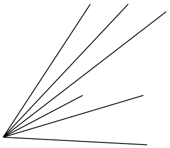
Modern cone algs have two main steps:

- Find some/all stable cones

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

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure



Cone basics

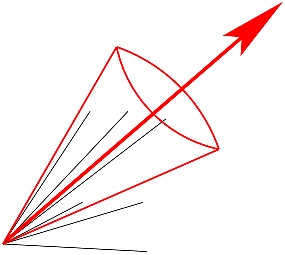
Modern cone algs have two main steps:

- Find some/all stable cones

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

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure



Cone basics

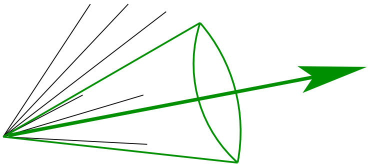
Modern cone algs have two main steps:

- Find some/all stable cones

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

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure



Cone basics

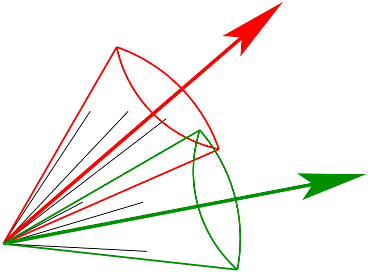
Modern cone algs have two main steps:

- Find some/all stable cones

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

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure



Cone basics

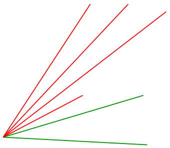
Modern cone algs have two main steps:

- Find some/all stable cones

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

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure



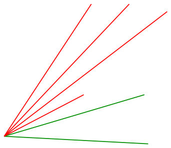
Cone basics

Modern cone algs have two main steps:

- Find some/all stable cones

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

  By running a 'split–merge' procedure



Qu: How do you find the stable cones?

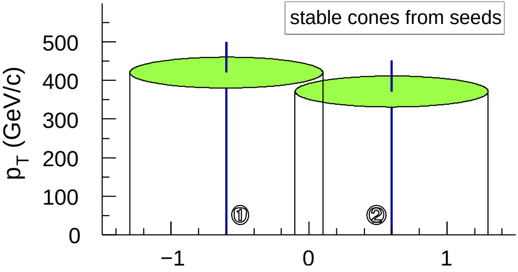
All experiments use iterative methods:

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

'Midpoint' algorithm

Cone theory issues

Use of seeds is *dangerous*



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

This is **IR unsafe**.
Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$

Kilgore & Giele '97

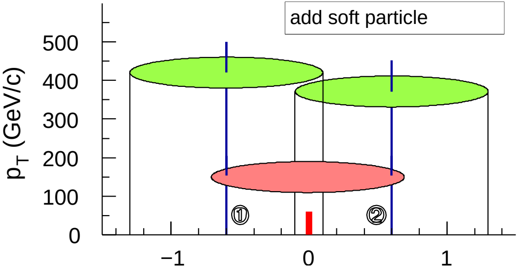
Solution: add extra seeds at midpoints of all pairs, triplets, ...of stable cones.

Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses...

Cone theory issues

Use of seeds is *dangerous*



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

This is **IR unsafe**.

Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$

Kilgore & Giele '97

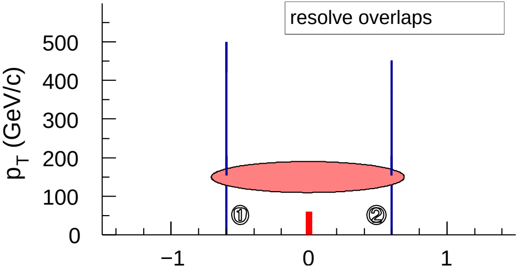
Solution: add extra seeds at midpoints of all pairs, triplets, ...of stable cones.

Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses...

Cone theory issues

Use of seeds is *dangerous*



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

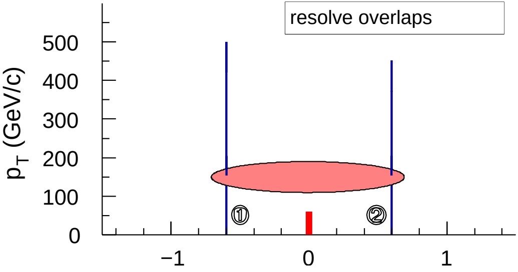
This is **IR unsafe**.
Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$
Kilgore & Giele '97

Solution: add extra seeds at midpoints of all pairs, triplets, . . . of stable cones.
Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses. . .

Cone theory issues

Use of seeds is *dangerous*



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

This is **IR unsafe**.
Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$

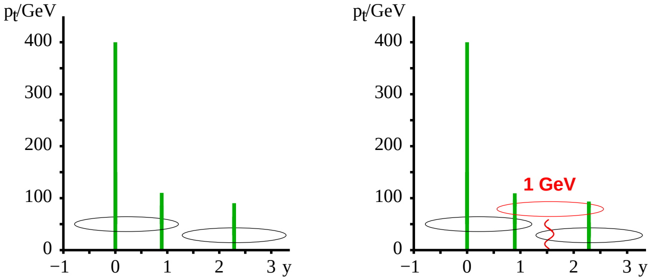
Kilgore & Giele '97

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

Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses. . .

Midpoint IR problem



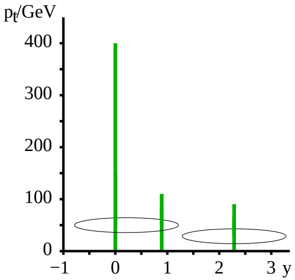
Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

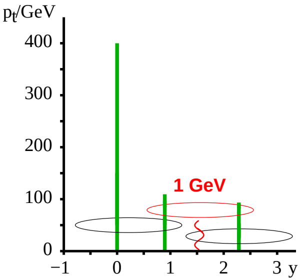
Midpoint cone alg. misses some stable cones; extra soft particle → extra starting point → extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR problem





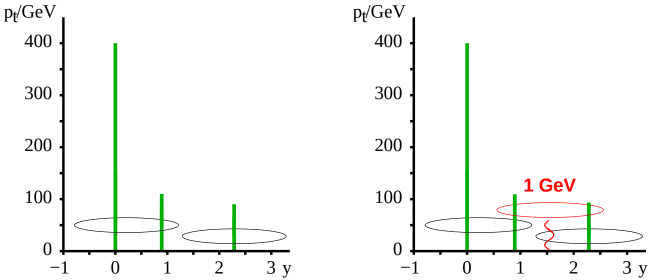
Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

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

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR problem



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

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

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR unsafety? Who cares?

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

IR/Collinear unsafety is a serious problem!

- Invalidates theorems that ensure finiteness of perturbative QCD

  Cancellation of real & virtual divergences
- Destroys usefulness of (intuitive) partonic picture

  you cannot think in terms of hard partons if adding a 1 GeV gluon changes 100 GeV jets
- 'Pragmatically:' limits accuracy to which it makes sense to calculate

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

$50 million worth of work for nothing?

Midpoint IR unsafety? Who cares?

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

IR/Collinear unsafety is a serious problem!

- Invalidates theorems that ensure finiteness of perturbative QCD

  Cancellation of real & virtual divergences

- Destroys usefulness of (intuitive) partonic picture

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

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

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

$50 million worth of work for nothing?

Seedless cone algorithms

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

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

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

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

One known exact approach:

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

Seedless cone algorithms

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

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

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

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

One known exact approach:

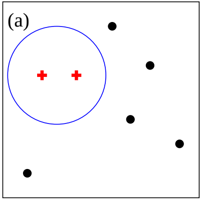
- Take each possible subset of particles and see if it forms a stable cone.
  Tevatron Run II workshop, '00 (for fixed-order calcs.)
- There are 2^N subsets for N particles. Computing time $\sim$ **$N2^N$**.
  10^{17} years for an event with 100 particles

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

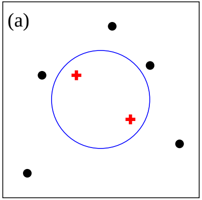
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

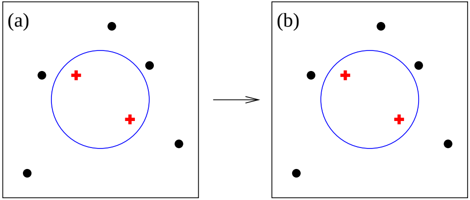
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

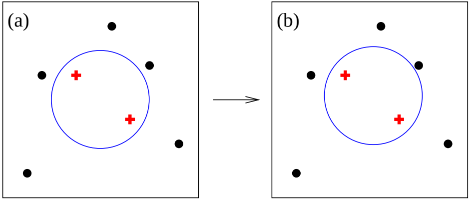
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

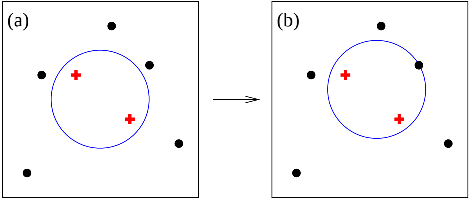
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

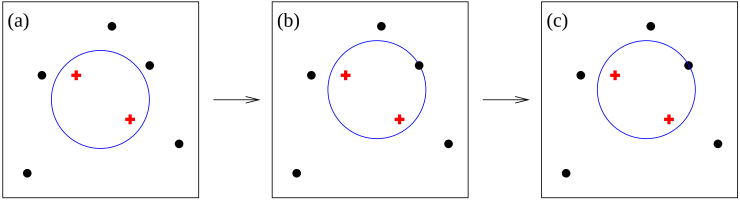
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

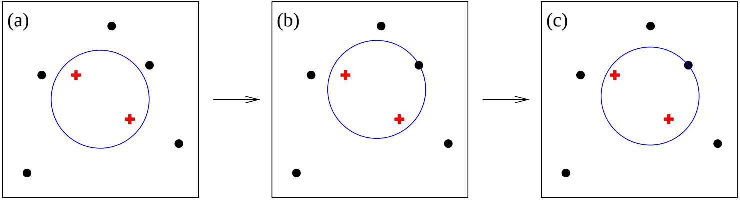
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

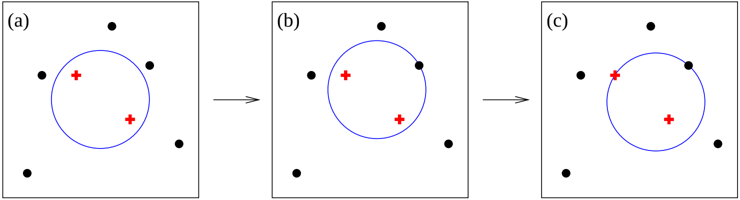
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

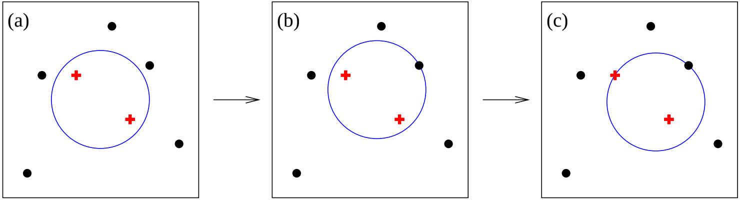
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

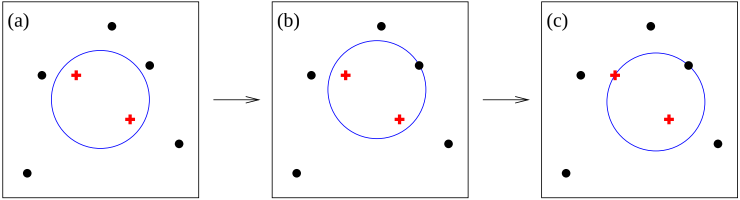
- For each *pair* of points in the plane, draw the two circles that have those two points on their edge.

Transform into a geometrical problem

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

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

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



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

Polynomial time recipe for finding all distinct enclosures:

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

A Seedless Infrared Safe Cone: SISCone

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

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

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

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

Traversal order, stability check

checkxor

GPS & Soyez '07

- Much faster than midpoint with no seed threshold

IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

Collinear unsafe

A Seedless Infrared Safe Cone: SISCone

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

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

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

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

Traversal order, stability check

checkxor

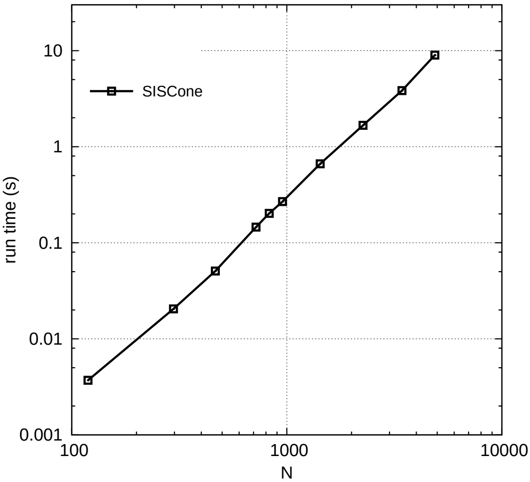
GPS & Soyez '07

- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

  Collinear unsafe



A Seedless Infrared Safe Cone: SISCone

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

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

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

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

Traversal order, stability check

checkxor

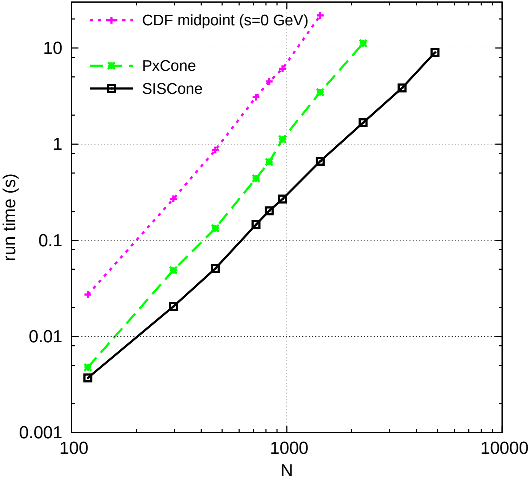
GPS & Soyez '07

- Much faster than midpoint with no seed threshold

IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

Collinear unsafe



A Seedless Infrared Safe Cone: SISCone

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

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

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

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

Traversal order, stability check

checkxor

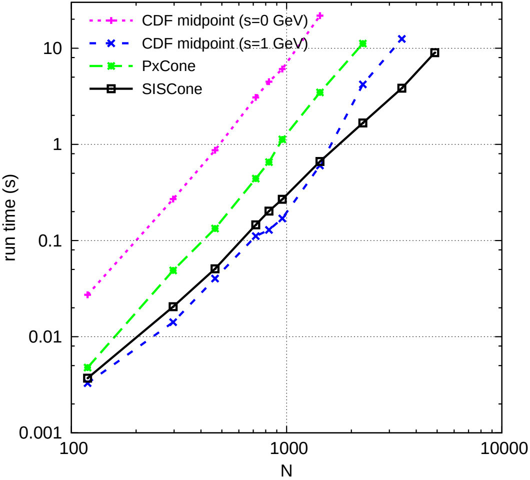
GPS & Soyez '07

- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

  Collinear unsafe



A Seedless Infrared Safe Cone: SISCone

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

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

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

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

Traversal order, stability check

checkxor

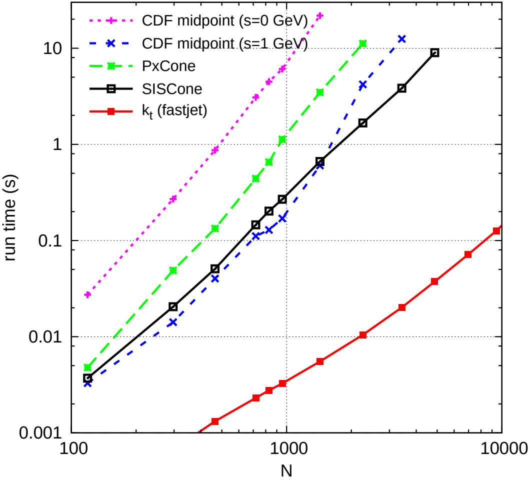
GPS & Soyez '07

- Much faster than midpoint with no seed threshold

  IR unsafe

- Same speed as midpoint codes with seeds > 1 GeV

  Collinear unsafe



MC cross check of IR safety

- 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 !

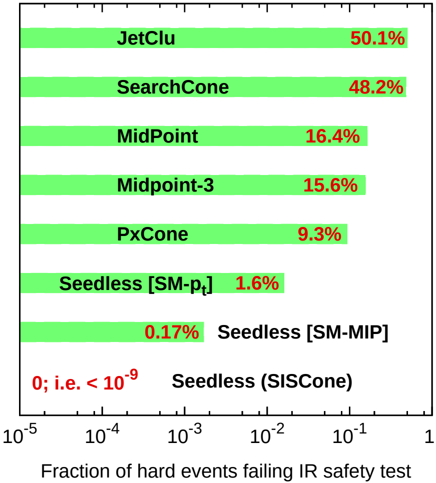
Be careful with split–merge too

MC cross check of IR safety

- 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



A full set of algs

Complementary set of IR/Collinear safe jet algs ⟶ flexbility in studying complex events.

Consider families of jet algs: e.g. sequential recombination with

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

	Alg. name	Comp. Geometry problem	time
$p = 1$	k_t CDOSTW '91-93; ES '93	Dynamic Nearest Neighbour CGAL (Devillers et al)	$N \ln N$ exp.
$p = 0$	Cambridge/Aachen Dok, Leder, Moretti, Webber '97 Wengler, Wobisch '98	Dynamic Closest Pair T. Chan '02	$N \ln N$
$p = -1$	anti-k_t (cone-like) Cacciari, GPS, Soyez, in prep.	Dynamic Nearest Neighbour CGAL (worst case)	$N^{3/2}$
cone	SISCone GPS Soyez '07 + Tevatron run II '00	All circular enclosures previously unconsidered	$N^2 \ln N$ exp.

All accessible in FastJet

FastJet in software of all (4) LHC collaborations

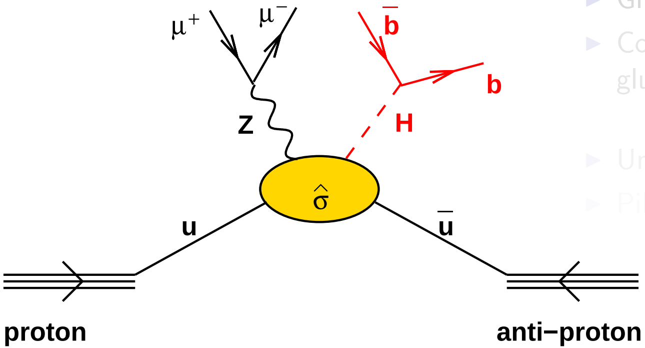
Once you have a decent set of jet algs, *start asking questions about them.*

- They share a common parameter R (angular reach). How do results depend on R?
- In what way do the various algorithms differ?
- How are they to be best used in the challenging LHC environment?

Try to answer questions with Monte Carlo? Gives little understanding of underlying principles.

➥*Supplement with analytical approximations.*

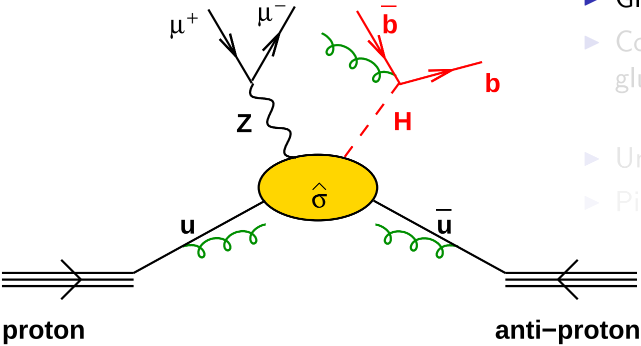
Various contributions



- Gluon emission, $\mathcal{O}(\alpha_s)$
- Conversion of quarks, gluons $\rightarrow \pi^{\pm}$, etc.

  Hadronisation
- Underlying event
- Pileup

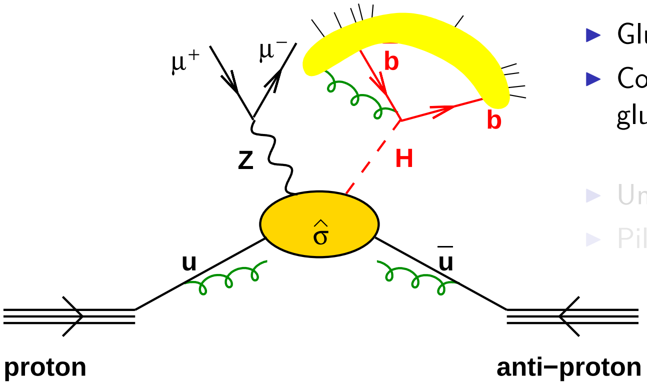
Various contributions



- Gluon emission, $\mathcal{O}(\alpha_s)$
- Conversion of quarks, gluons $\rightarrow \pi^{\pm}$, etc.

  Hadronisation
- Underlying event
- Pileup

Various contributions

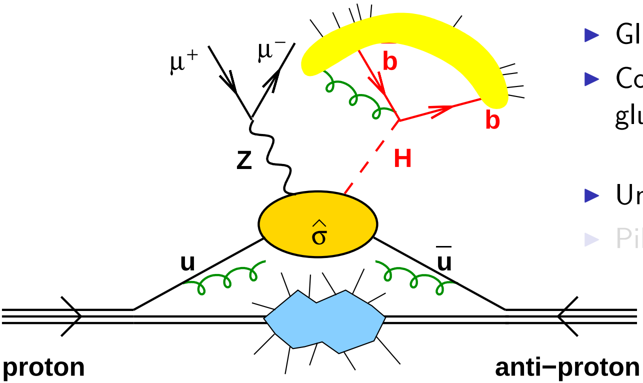


- Gluon emission, $\mathcal{O}(\alpha_s)$
- Conversion of quarks, gluons → $\pi^{\pm}$, etc.

Hadronisation

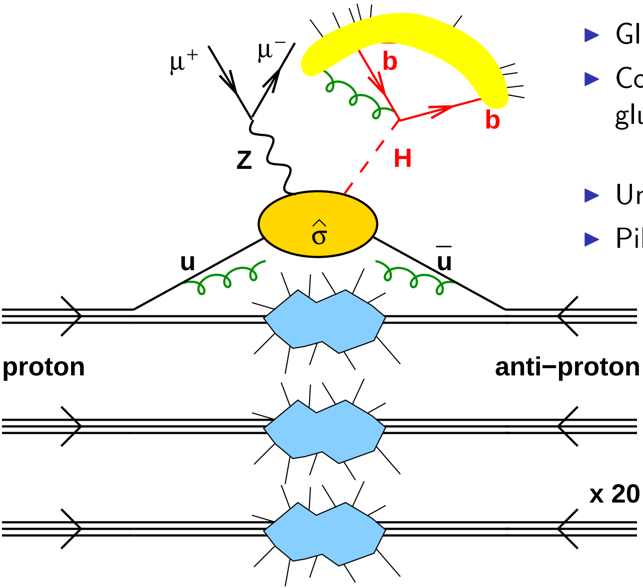
- Underlying event
- Pileup

Various contributions



- Gluon emission, $\mathcal{O}(\alpha_s)$
- Conversion of quarks, gluons $\to \pi^\pm$, etc.
  Hadronisation
- Underlying event
- Pileup

Various contributions



- Gluon emission, $\mathcal{O}(\alpha_s)$
- Conversion of quarks, gluons $\to \pi^{\pm}$, etc.

  Hadronisation
- Underlying event
- Pileup

Perturbative

Start with *quark* with transverse momentum p_t

$$\langle \delta p_t \rangle_{PT} \simeq \frac{1}{\sigma_0} \int d\Phi |M^2| \, \alpha_s(k_{t,rel}) \, (p_{t,jet} - p_t)$$

$$\simeq \frac{\alpha_s C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \cdot ((1-z)p_t - p_t)$$

$$\simeq -1.01 \frac{\alpha_s C_F}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t) \qquad C_F = 4/3$$

Similarly for gluon:

$$\langle \delta p_t \rangle_{PT} \simeq -(0.94 C_A + 0.15 n_f T_R) \frac{\alpha_s}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$$C_A = 3$$

NB1: $\alpha_s p_t \ln R$ structure & coeff. independent of process
NB2: these and subsequent results hold for *all algorithms* (1-gluon approx).

Perturbative

Start with *quark* with transverse momentum p_t

$$\langle \delta p_t \rangle_{PT} \simeq \frac{1}{\sigma_0} \int d\Phi |M^2| \, \alpha_s(k_{t,rel}) \, (p_{t,jet} - p_t)$$

$$\simeq \frac{\alpha_s C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \cdot ((1-z)p_t - p_t)$$

$$\simeq -1.01 \frac{\alpha_s C_F}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t) \qquad C_F = 4/3$$

Similarly for gluon:

$$\langle \delta p_t \rangle_{PT} \simeq -(0.94 C_A + 0.15 n_f T_R) \frac{\alpha_s}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t) \qquad C_A = 3$$

NB1: $\alpha_s p_t \ln R$ structure & coeff. independent of process

NB2: these and subsequent results hold for *all algorithms* (1-gluon approx).

Perturbative

Start with *quark* with transverse momentum p_t

$$\langle \delta p_t \rangle_{PT} \simeq \frac{1}{\sigma_0} \int d\Phi |M^2| \, \alpha_s(k_{t,rel}) \, (p_{t,jet} - p_t)$$

$$\simeq \frac{\alpha_s C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \cdot (-z p_t)$$

$$\simeq -1.01 \frac{\alpha_s C_F}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_F = 4/3$

Similarly for gluon:

$$\langle \delta p_t \rangle_{PT} \simeq -(0.94 C_A + 0.15 n_f T_R) \frac{\alpha_s}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_A = 3$

NB1: $\alpha_s p_t \ln R$ structure & coeff. independent of process
NB2: these and subsequent results hold for *all algorithms* (1-gluon approx).

Perturbative

Start with *quark* with transverse momentum p_t

$$\langle \delta p_t \rangle_{PT} \simeq \frac{1}{\sigma_0} \int d\Phi |M^2| \, \alpha_s(k_{t,rel}) \, (p_{t,jet} - p_t)$$

$$\simeq \frac{\alpha_s C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \cdot (-z p_t)$$

$$\simeq -1.01 \frac{\alpha_s C_F}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_F = 4/3$

Similarly for gluon:

$$\langle \delta p_t \rangle_{PT} \simeq -(0.94 C_A + 0.15 n_f T_R) \frac{\alpha_s}{\pi} p_t \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_A = 3$

NB1: $\alpha_s p_t \ln R$ structure & coeff. independent of process
NB2: these and subsequent results hold for *all algorithms* (1-gluon approx).

Perturbative

Start with *quark* with transverse momentum p_t

$$\langle \delta p_t \rangle_{PT} \simeq \frac{1}{\sigma_0} \int d\Phi |M^2| \, \alpha_s(k_{t,rel}) \, (p_{t,jet} - p_t)$$

$$\simeq \frac{\alpha_s C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \cdot (-z p_t)$$

$$\simeq -1.01 \frac{\alpha_s C_F}{\pi} \, p_t \, \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_F = 4/3$

Similarly for gluon:

$$\langle \delta p_t \rangle_{PT} \simeq -\left(0.94 C_A + 0.15 n_f T_R\right) \frac{\alpha_s}{\pi} \, p_t \, \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_A = 3$

NB1: $\alpha_s p_t \ln R$ structure & coeff. independent of process
NB2: these and subsequent results hold for *all algorithms* (1-gluon approx).

Perturbative

Start with *quark* with transverse momentum p_t

$$\langle \delta p_t \rangle_{PT} \simeq \frac{1}{\sigma_0} \int d\Phi |M^2| \, \alpha_s(k_{t,rel}) \, (p_{t,jet} - p_t)$$

$$\simeq \frac{\alpha_s C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \cdot (-z p_t)$$

$$\simeq -1.01 \frac{\alpha_s C_F}{\pi} \, p_t \, \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_F = 4/3$

Similarly for gluon:

$$\langle \delta p_t \rangle_{PT} \simeq -(0.94 C_A + 0.15 n_f T_R) \frac{\alpha_s}{\pi} \, p_t \, \ln \frac{1}{R} + \mathcal{O}(\alpha_s p_t)$$

$C_A = 3$

NB1: $\alpha_s p_t \ln R$ structure & coeff. independent of process
NB2: these and subsequent results hold for *all algorithms* (1-gluon approx).

Hadronisation

Simplest form of a trick developed $\sim$ 1995: to establish non-perturbative contribution, replace $\alpha_s(k_{t,rel}) \to \delta\alpha_s(k_{t,rel})$, with support only near Λ_{QCD}.

Dokshitzer & Webber; Korchemsky & Sterman
Akhoury & Zakharov; Beneke & Braun

E.g.:

$$\frac{2}{\pi}\delta\alpha_s(k_{t,rel}) = \Lambda\delta(k_{t,rel} - \Lambda)$$

$\Lambda = \int dk_{t,rel}\delta\alpha_s(k_{t,rel})$, should be **‘universal’**.

Tested for $\sim$ 10 observables in e^+e^- and DIS.

$\alpha_0 \simeq 0.5 \leftrightarrow \Lambda \simeq 0.4$ GeV

Hadronisation

Simplest form of a trick developed $\sim$ 1995: to establish non-perturbative contribution, replace $\alpha_s(k_{t,rel}) \to \delta\alpha_s(k_{t,rel})$, with support only near Λ_{QCD}.

Dokshitzer & Webber; Korchemsky & Sterman
Akhoury & Zakharov; Beneke & Braun

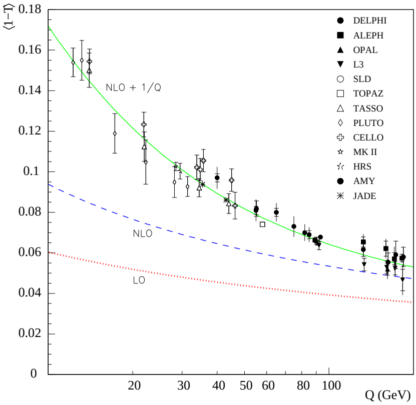
E.g.:

$$\frac{2}{\pi}\delta\alpha_s(k_{t,rel}) = \Lambda\delta(k_{t,rel} - \Lambda)$$

$\Lambda = \int dk_{t,rel}\delta\alpha_s(k_{t,rel})$, should be **'universal'**.

Tested for $\sim$ 10 observables in e^+e^- and DIS.

$\alpha_0 \simeq 0.5 \leftrightarrow \Lambda \simeq 0.4$ GeV



Hadronisation

Simplest form of a trick developed $\sim$ 1995: to establish non-perturbative contribution, replace $\alpha_s(k_{t,rel}) \to \delta\alpha_s(k_{t,rel})$, with support only near Λ_{QCD}.

Dokshitzer & Webber; Korchemsky & Sterman
Akhoury & Zakharov; Beneke & Braun

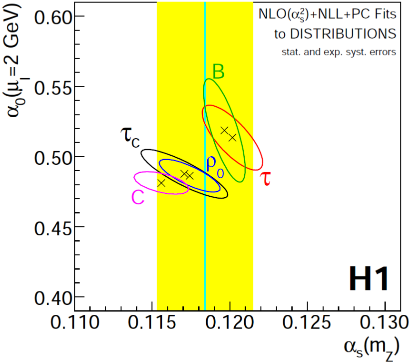
E.g.:

$$\frac{2}{\pi}\delta\alpha_s(k_{t,rel}) = \Lambda\delta(k_{t,rel} - \Lambda)$$

$\Lambda = \int dk_{t,rel}\delta\alpha_s(k_{t,rel})$, should be **'universal'**.

Tested for $\sim$ 10 observables in e^+e^- and DIS.

$\alpha_0 \simeq 0.5 \leftrightarrow \Lambda \simeq 0.4$ GeV



H1 data; Dasgupta & GPS '02

Hadronisation (cont.)

Hadronisation for quarks:

$$\langle \delta p_t \rangle_{hadr} \simeq \frac{C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \, \delta\alpha_s(z\theta p_t) \cdot (-z p_t)$$

$$= -\frac{C_F \Lambda}{R} + \mathcal{O}(\Lambda R) \qquad \text{gluons:} \quad -\frac{C_A \Lambda}{R}$$

Dasgupta, Magnea & GPS '07
Deducible from Korchemsky & Sterman '94
Seymour '97; but lost in mists of time.

If underlying event had similar mechanism, we'd get:

$$\langle \delta p_t \rangle_{UE} \simeq \frac{2C_F}{\pi} \int_R^{\mathcal{O}(1)} \theta d\theta \int dz \, \frac{dk_t}{k_t} \, \delta\alpha_s(k_t) \cdot (k_t)$$

$$\simeq C_F \Lambda \frac{R^2}{2} + \mathcal{O}(\Lambda R^4)$$

NB: to first approx., all jet algorithms identical

Hadronisation (cont.)

Hadronisation for quarks:

$$\langle \delta p_t \rangle_{hadr} \simeq \frac{C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz\, p_{gq}(z)\, \delta\alpha_s(z\theta p_t) \cdot (-z p_t)$$

$$= -\frac{C_F \Lambda}{R} + \mathcal{O}(\Lambda R) \qquad \text{gluons:} \quad -\frac{C_A \Lambda}{R}$$

Dasgupta, Magnea & GPS '07
Deducible from Korchemsky & Sterman '94
Seymour '97; but lost in mists of time.

If underlying event had similar mechanism, we'd get:

$$\langle \delta p_t \rangle_{UE} \simeq \frac{2C_F}{\pi} \int_R^{\mathcal{O}(1)} \theta d\theta \int dz \frac{dk_t}{k_t} \delta\alpha_s(k_t) \cdot (k_t)$$

$$\simeq C_F \Lambda \frac{R^2}{2} + \mathcal{O}(\Lambda R^4)$$

NB: to first approx., all jet algorithms identical

Hadronisation (cont.)

Hadronisation for quarks:

$$\langle \delta p_t \rangle_{hadr} \simeq \frac{C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz\, p_{gq}(z)\, \delta\alpha_s(z\theta p_t) \cdot (-z p_t)$$

$$= -\frac{C_F \Lambda}{R} + \mathcal{O}(\Lambda R) \qquad \text{gluons:} \quad -\frac{C_A \Lambda}{R}$$

Dasgupta, Magnea & GPS '07

Deducible from Korchemsky & Sterman '94

Seymour '97; but lost in mists of time.

If underlying event had similar mechanism, we'd get:

$$\langle \delta p_t \rangle_{UE} \simeq \frac{2C_F}{\pi} \int_R^{\mathcal{O}(1)} \theta\, d\theta \int dz\, \frac{dk_t}{k_t}\, \delta\alpha_s(k_t) \cdot (k_t)$$

$$\simeq C_F \Lambda \frac{R^2}{2} + \mathcal{O}(\Lambda R^4)$$

NB: to first approx., all jet algorithms identical

Hadronisation (cont.)

Hadronisation for quarks:

$$\langle \delta p_t \rangle_{hadr} \simeq \frac{C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz \, p_{gq}(z) \, \delta\alpha_s(z\theta p_t) \cdot (-z p_t)$$

$$= -\frac{C_F \Lambda}{R} + \mathcal{O}(\Lambda R) \qquad \text{gluons:} \quad -\frac{C_A \Lambda}{R}$$

Dasgupta, Magnea & GPS '07
Deducible from Korchemsky & Sterman '94
Seymour '97; but lost in mists of time.

If underlying event had similar mechanism, we'd get:

$$\langle \delta p_t \rangle_{UE} \simeq \frac{2C_F}{\pi} \int_R^{\mathcal{O}(1)} \theta d\theta \int dz \, \frac{dk_t}{k_t} \, \delta\alpha_s(k_t) \cdot (k_t)$$

$$\simeq C_F \Lambda \frac{R^2}{2} + \mathcal{O}(\Lambda R^4)$$

NB: to first approx., all jet algorithms identical

Hadronisation (cont.)

Hadronisation for quarks:

$$\langle \delta p_t \rangle_{hadr} \simeq \frac{C_F}{\pi} \int_R^{\mathcal{O}(1)} \frac{d\theta}{\theta} \int dz\, p_{gq}(z)\, \delta\alpha_s(z\theta p_t) \cdot (-z p_t)$$

$$= -\frac{C_F \Lambda}{R} + \mathcal{O}(\Lambda R) \qquad \text{gluons:} \quad -\frac{C_A \Lambda}{R}$$

Dasgupta, Magnea & GPS '07
Deducible from Korchemsky & Sterman '94
Seymour '97; but lost in mists of time.

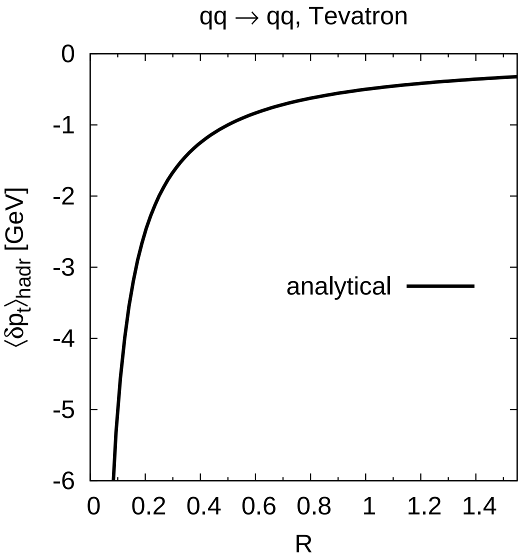
If underlying event had similar mechanism, we'd get:

$$\langle \delta p_t \rangle_{UE} \simeq \frac{2C_F}{\pi} \int_R^{\mathcal{O}(1)} \theta d\theta \int dz\, \frac{dk_t}{k_t}\, \delta\alpha_s(k_t) \cdot (k_t)$$

$$\simeq C_F \Lambda \frac{R^2}{2} + \mathcal{O}(\Lambda R^4)$$

NB: to first approx., all jet algorithms identical

Test NP results v. MC



MC hadr. agrees with calc.

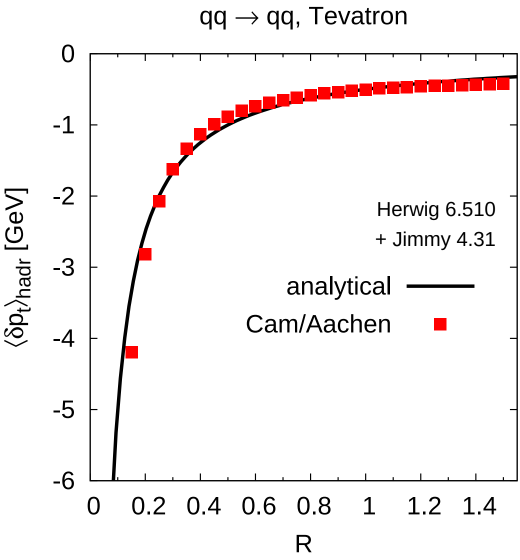
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE ≫ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is ∼ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

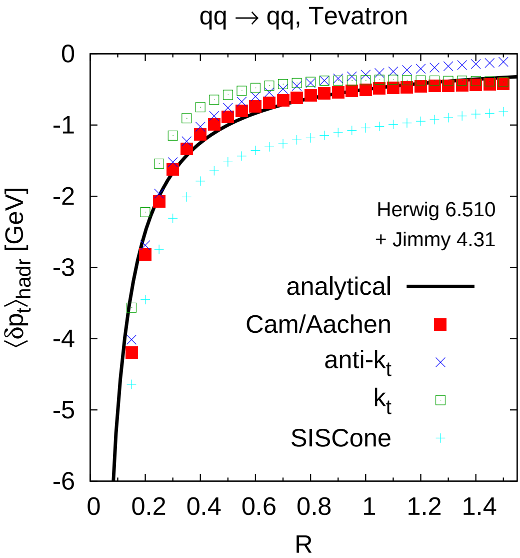
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE ≫ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is ∼ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

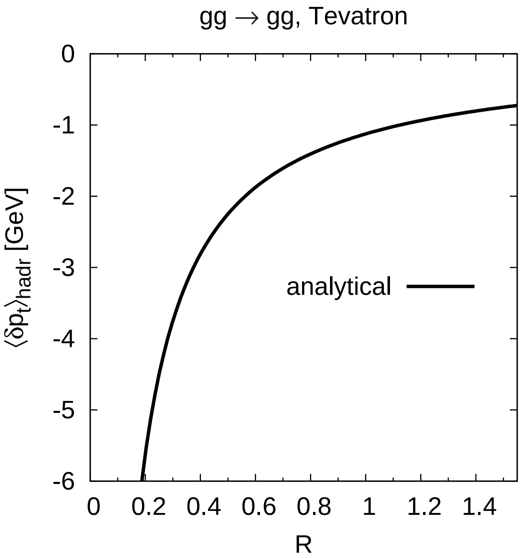
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE $\gg$ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is $\sim$ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

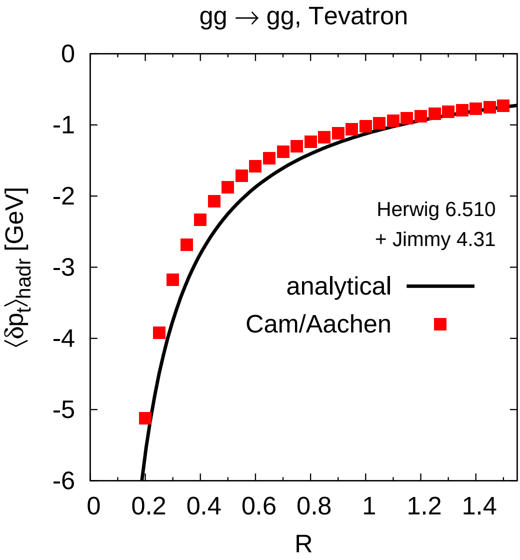
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE $\gg$ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is $\sim$ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

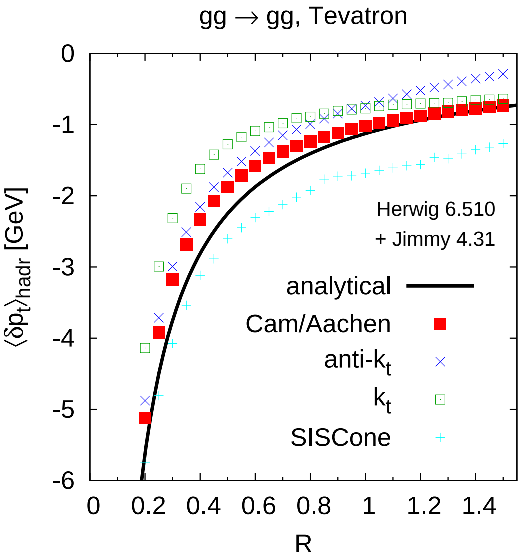
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE ≫ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is ∼ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

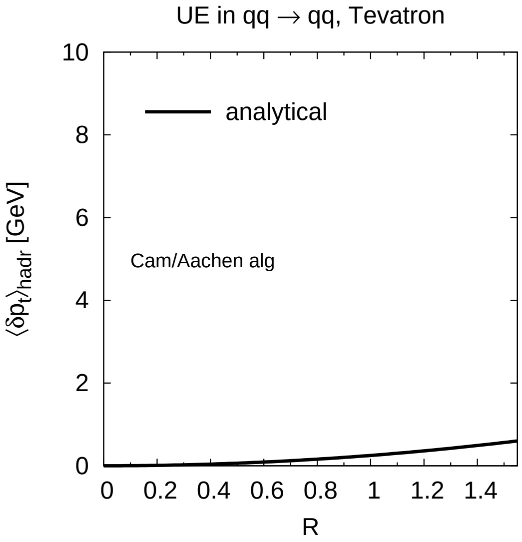
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE ≫ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is ∼ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

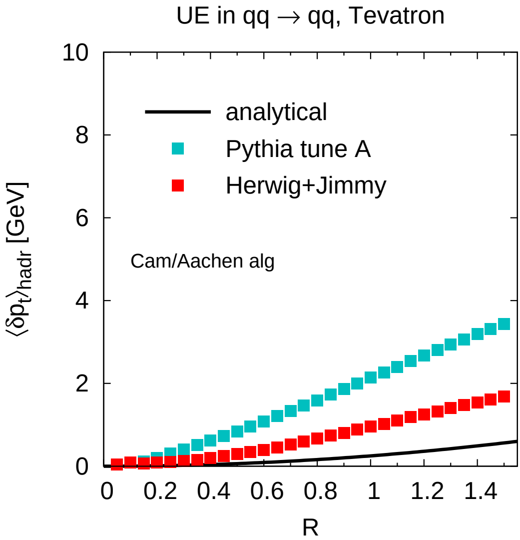
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE $\gg$ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is $\sim$ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

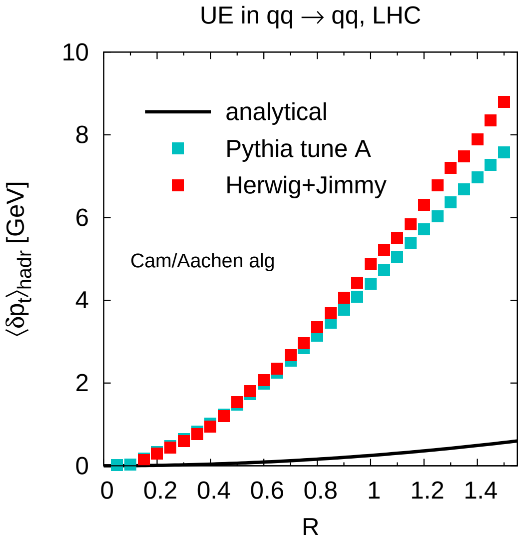
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE $\gg$ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is $\sim$ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

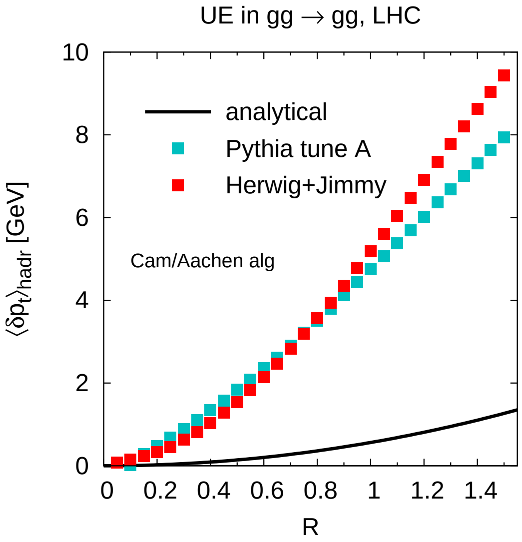
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE $\gg$ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is $\sim$ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

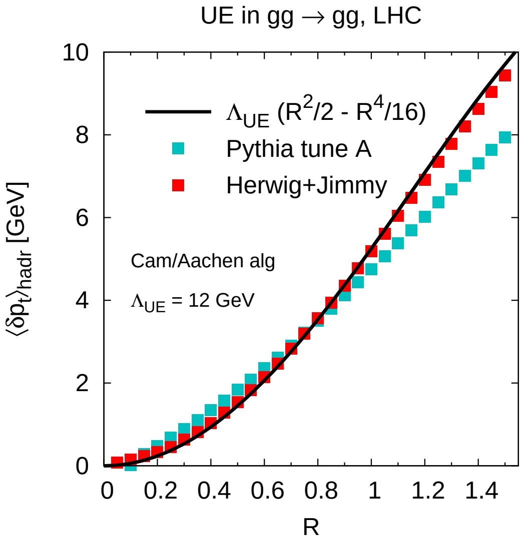
- to varying degrees for range of algs
- also in larger gluonic channels

MC UE $\gg$ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is $\sim$ 10 GeV

Test NP results v. MC



MC hadr. agrees with calc.

- to varying degrees for range of algs
- also in larger gluonic channels

MC UE $\gg$ naive expectation

- models tuned on same data behave differently
- UE is huge at LHC
- largely indep. of scattering channel

Scale for (non-perturbative!) UE is $\sim$ 10 GeV

Optimal R?

	Jet $\langle \delta p_t \rangle$ given by product of dependence on			
	scale	colour factor	R	$\sqrt{s}$
pert. radiation	$\sim \frac{\alpha_s(p_t)}{\pi} p_t$	C_i	$\ln R + \mathcal{O}(1)$	–
hadronisation	Λ_h	C_i	$-1/R + \mathcal{O}(R)$	–
UE	Λ_{UE}	–	$R^2/2 + \mathcal{O}(R^4)$	s^{ω}

To get best experimental resolutions, minimise contributions from all 3 components.

Here: sum of squared means

Better still: calculate flucts

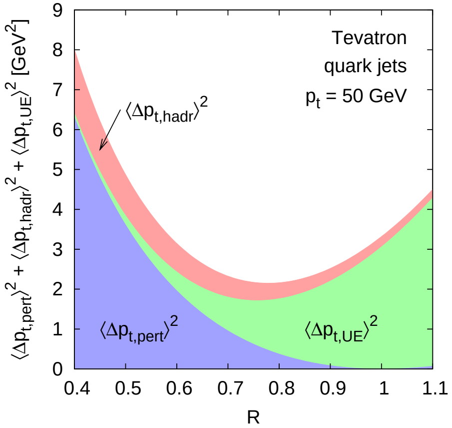
NB: this is rough picture

details of p_t scaling wrong

But can still be used to understand general principles.

Optimal R?

	Jet $\langle \delta p_t \rangle$ given by product of dependence on			
	scale	colour factor	R	$\sqrt{s}$
pert. radiation	$\sim \frac{\alpha_s(p_t)}{\pi}\, p_t$	C_i	$\ln R + \mathcal{O}(1)$	–
hadronisation	Λ_h	C_i	$-1/R + \mathcal{O}(R)$	–
UE	Λ_{UE}	–	$R^2/2 + \mathcal{O}(R^4)$	s^{ω}



To get best experimental resolutions, minimise contributions from all 3 components.

Here: sum of squared means

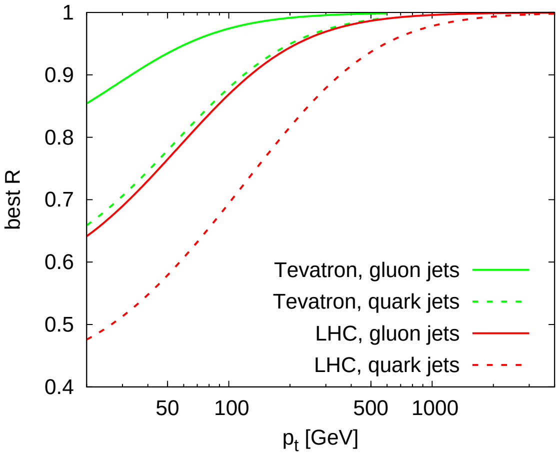
Better still: calculate flucts

NB: this is rough picture

details of p_t scaling wrong

But can still be used to understand general principles.

Optimal R v p_t, proc., collider

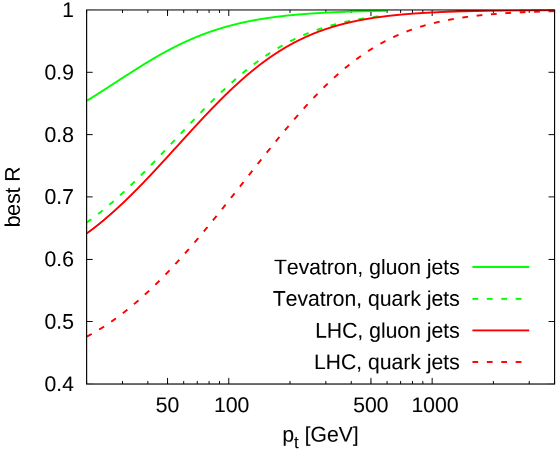


Basic messages

- higher $p_t \to$ larger R
  Most say opposite
- larger R for gluons than quarks   Gluon jets wider
- smaller R at LHC than Tevatron   UE larger

This kind of information is the start of what might go into
"auto-focus for jetography"

Optimal R v p_t, proc., collider



Basic messages

- higher $p_t \longrightarrow$ larger R
  Most say opposite
- larger R for gluons than quarks
  Gluon jets wider
- smaller R at LHC than Tevatron
  UE larger

This kind of information is the start of what might go into **"auto-focus for jetography"**

Other results

This last part of talk was an overview of *1 of several* recent jet topics

Others include

- Subtraction of pileup — Cacciari & GPS '07
- Jet areas ↔ sensitivity to UE/pileup — Cacciari, GPS & Soyez prelim
- "Optimising R" — cross checking with MC — Cacciari, Rojo, GPS & Soyez, for Les Houches
- Jet flavour — e.g. reducing b-jet theory uncertainties from $40-60\%$ to $10-20\%$. — Banfi, GPS & Zanderighi '06, '07

Conclusions / Outlook

- Jets are the closest we can get to seeing and giving meaning to partons
- Play a pivotal role in experimental analyses, comparisons to QCD calculations
- Significant progress in past 2 years towards making them *consistent* (IR/Collinear safe) and *practical*

  Link with computational geometry

  All tools are made public:

  `http://www.lpthe.jussieu.fr/~salam/fastjet/`
- The physics of how jets behave in a hadron-collider environment is a rich subject — much to be understood, and potential for significant impact in how jets are used at LHC