Jet Phenomenology and the KtJet Algorithm

Will Plano

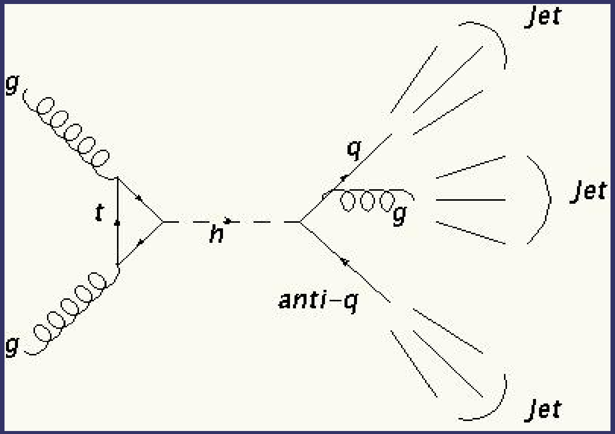
University of Manchester

Overview

- What are jets?
- The PXCONE algorithm
- Jets at the LHC
- The KtJet algorithm
- Coming soon...

What Are Jets?

- At parton level a jet is defined by a tree level quark or gluon from the hard subprocess in question, theoretically a nice description but experimentally impossible to measure.



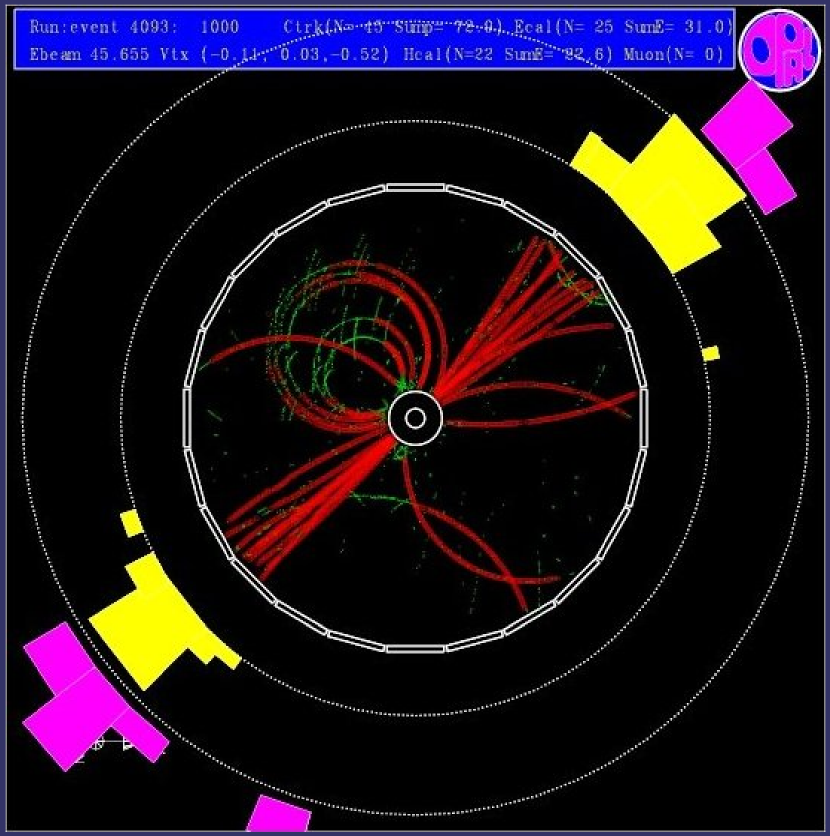
Example parton level jet definition in g + g -> Higgs -> q + qbar + g

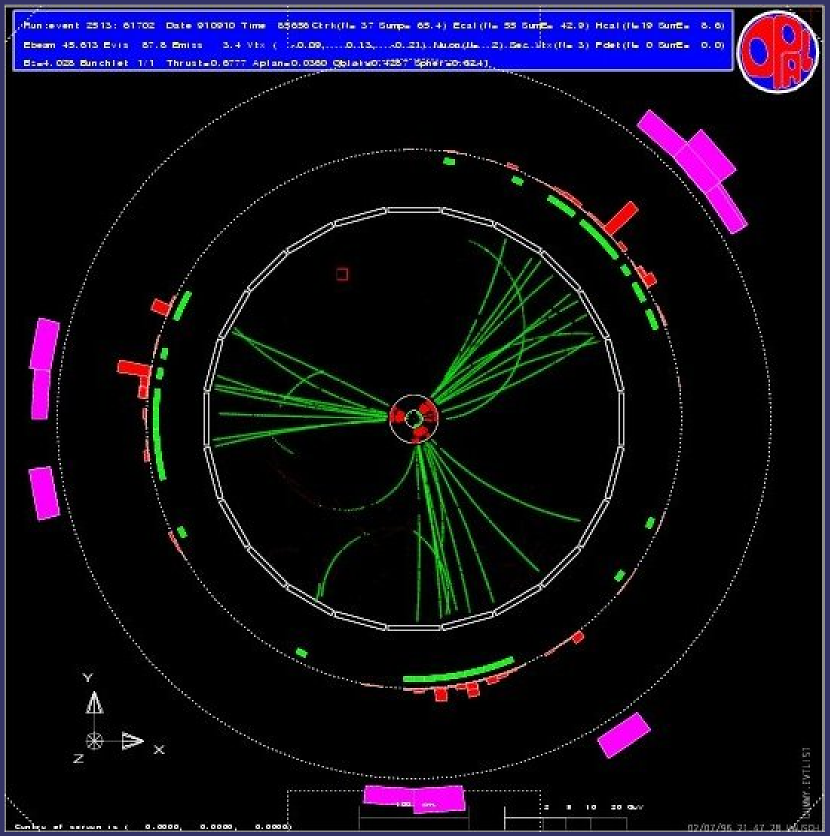
- Wikipedia:

  "A Jet is a narrow cone of hadrons and other particles..."

  This is a reasonable hadron level description of low energy jets, experimentally observable but difficult to work with mathematically.

2-Jet & 3-Jet Production at LEP





$e^+ + e^- \rightarrow Z^0 \rightarrow q + \text{qbar}$ $\qquad$ $e^+ + e^- \rightarrow Z^0 \rightarrow q + \text{qbar} + g$

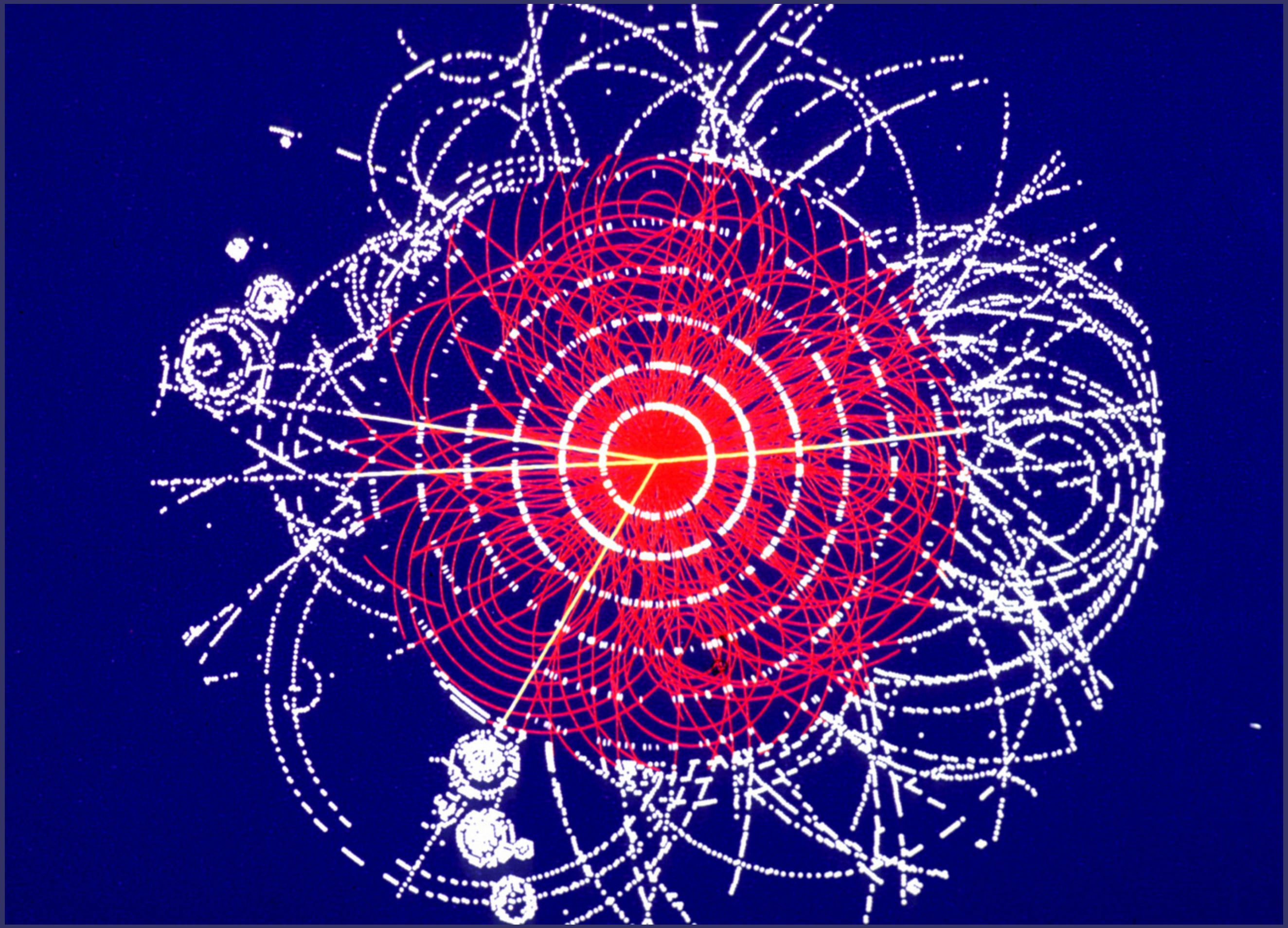
The PXCONE Algorithm

- PXCONE finds jets by moving cones of a given radius around in eta-phi space and then performing some clever tricks to ensure infra-red safety.
- Need to tune the radius parameter to the current analysis to get efficient jet finding with maximum background rejection.

Jets at the LHC

- At high luminosity (10^{34} $cm^{-2}s^{-1}$) at the LHC there will be on average around 20 minimum bias events per bunch crossing, that is 20 interactions overlaid on top of each other.
- This creates a problem for jet finding algorithms as jets will tend to include unwanted background “stuff”, this effect can be reduced if we can model these background events well.
- Another, perhaps more serious, problem is that at the high center of mass energy of the LHC (14 TeV) jets tend to have a larger intrinsic spread in angle. This means that jets tend to overlap more causing a problem for the cone based approach.

A Simulated LHC Event



The KtJet Algorithm

- KtJet attempts to provide an alternate solution to jet finding.
- It is based on a user definable (don't worry there are a few useful ones built in) separation parameter.
- This is calculated for each pair of objects and for each object and the beam.
- It then merges unwanted soft beam objects with the beam jets and discards them.
- Offers a choice of recombination schemes (ways in which objects are combined into jets)
- Can also give information on merging scales.

Coming Soon... KtJet 2.0!

- Will include a choice of either Kt or Cone based algorithm allowing very easy comparisons without having to write any extra code.
- Ability to use KtJet's recombination schemes in the cone algorithm.
- Now faster than ever. ^_^
- All in lovely objectified C++ for ease of use and extra modifiability. Why not try to write your own distance parameter?
- See it soon on http://hepforge.cedar.ac.uk/ktjet/
- But if you really can't wait or want more info, contact:
  - ktjet@cedar.ac.uk
- Or mail me directly at:
  - wplano@cern.ch