



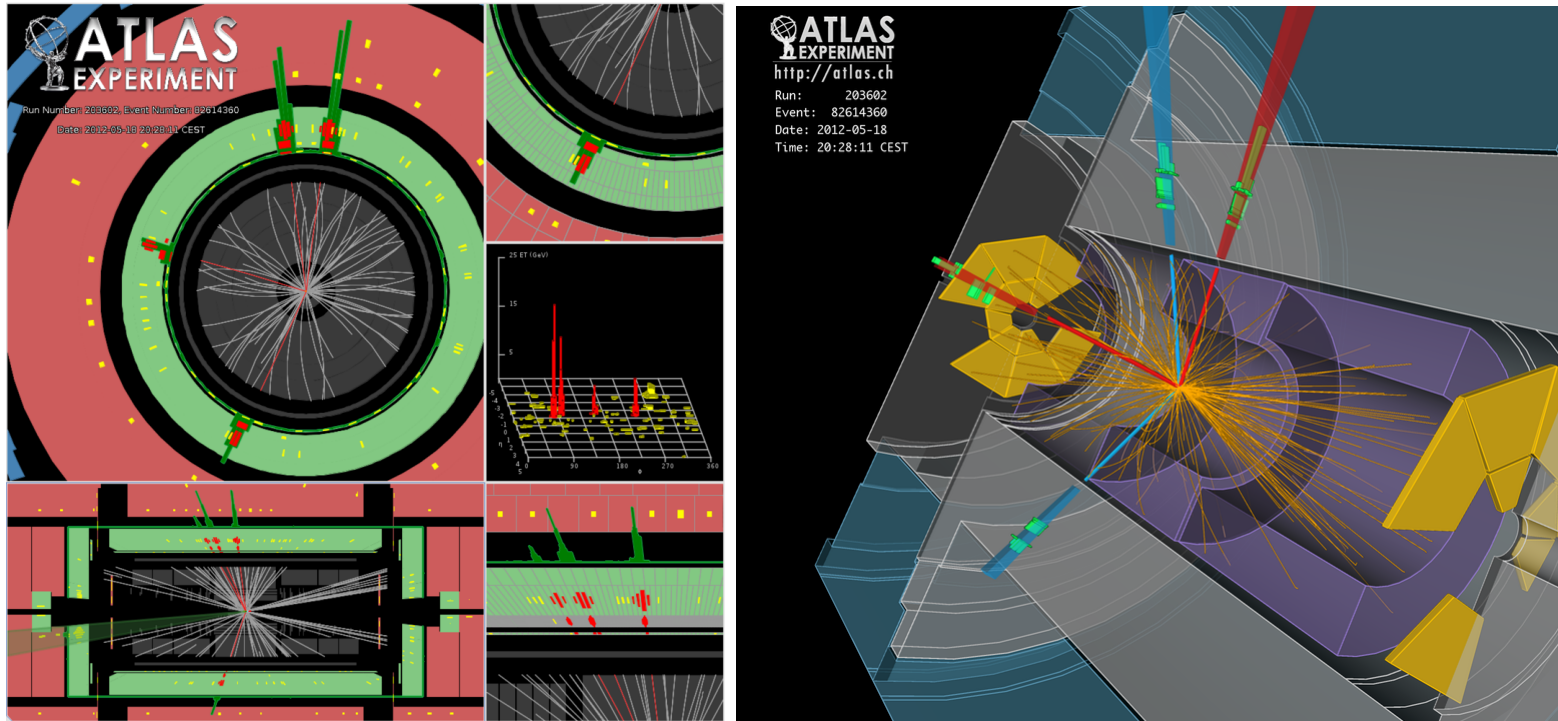
The Atlantis event display application

HSF Visualization workshop, 28th to 30th March 2017
<https://indico.cern.ch/event/617054/>

Visualising events

- Why do we need pictures?
 - visual investigation and the understanding of the physics of complete events
 - help develop reconstruction and analysis algorithms
 - facilitate debugging during commissioning
 - create pictures for publications, presentations and exhibitions
- Approaches
 - 3D vs 2D
 - data-oriented projections

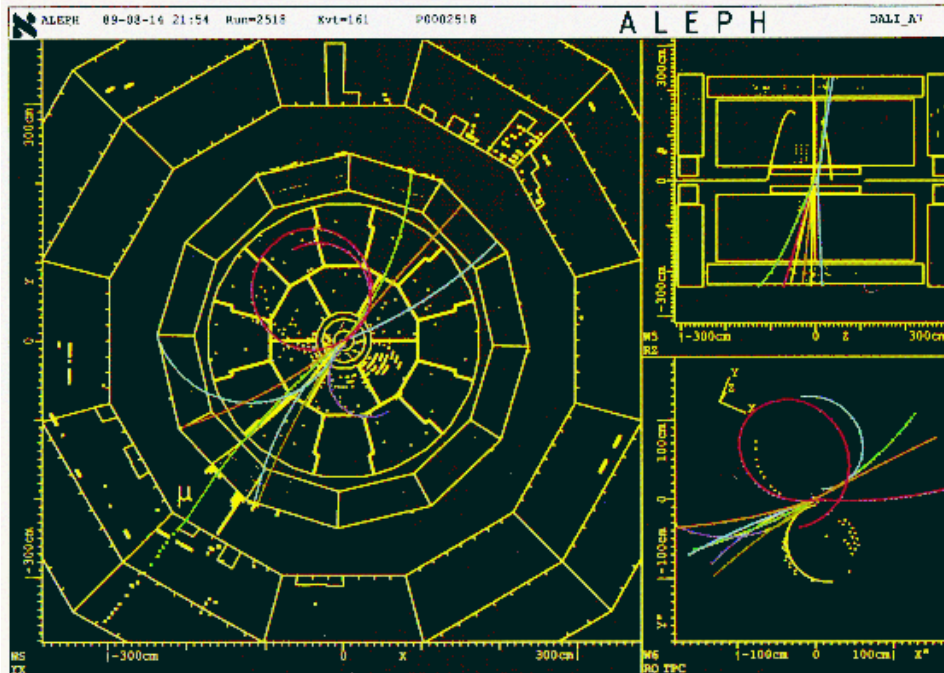
Atlantis / Virtual Point 1 (VP1) comparison



Atlantis

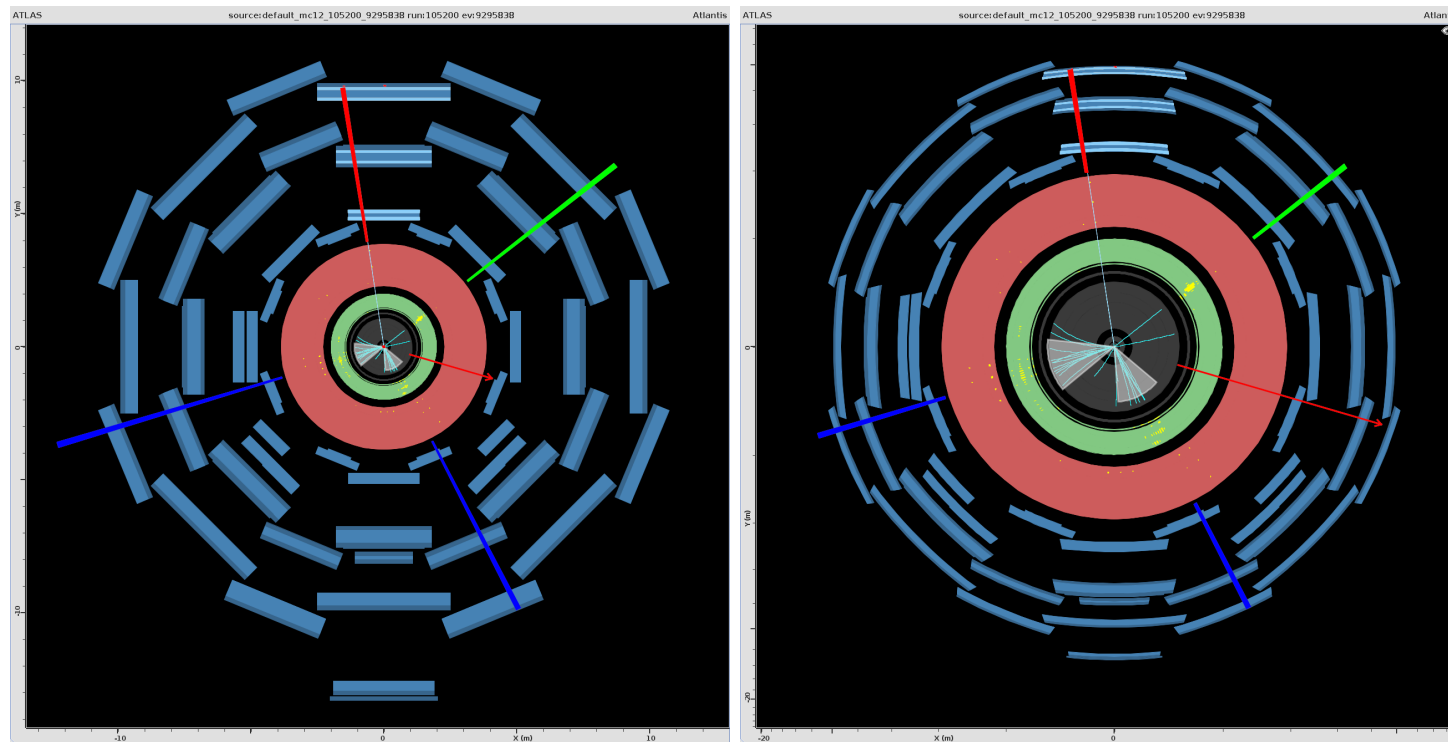
- Mostly 2D
- Hide the detector details in favour of clarity for the event data
- *Data-oriented projections*
 - use natural variables for pp collisions: rho, phi, eta
 - fish-eye projection
 - V-plot gives a lot of information about tracks
- Data in JiveXML format written by package in C++ framework

Origins of Atlantis



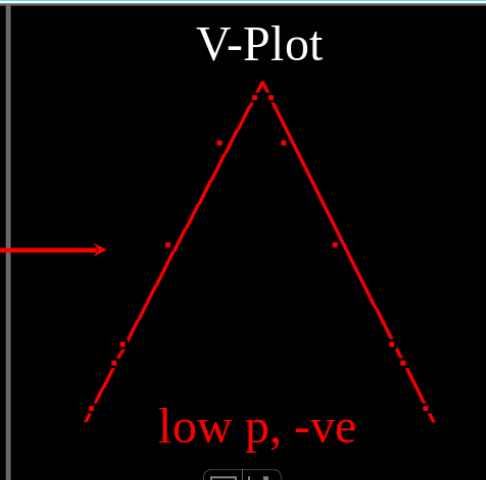
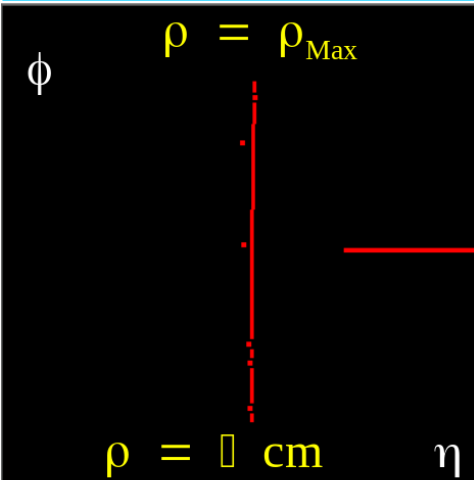
- DALI package written by Hans Drevermann for the ALEPH experiment
- Fortran 77
- Adapted by Hans for ATLAS
- Rewritten in Java at UC Santa Cruz

The fisheye projection



A data-oriented projection: the "V-Plot"

ATLAS Atlantis Event: test_event.xml



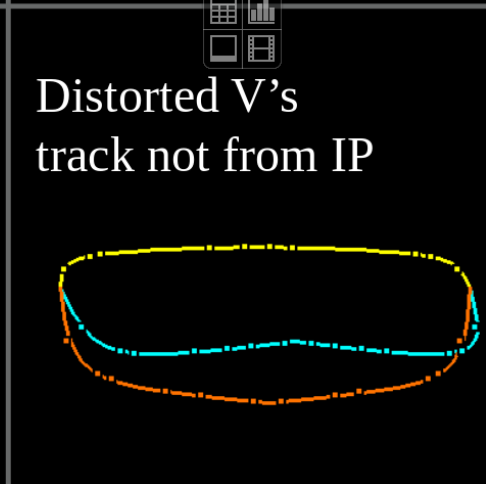
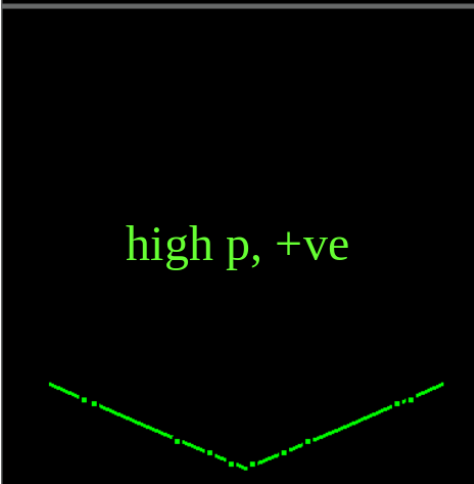
Draw each space point twice at

$\Phi, \eta + k^*(\rho - \rho_{Max})$

and

$\Phi, \eta - k^*(\rho - \rho_{Max})$

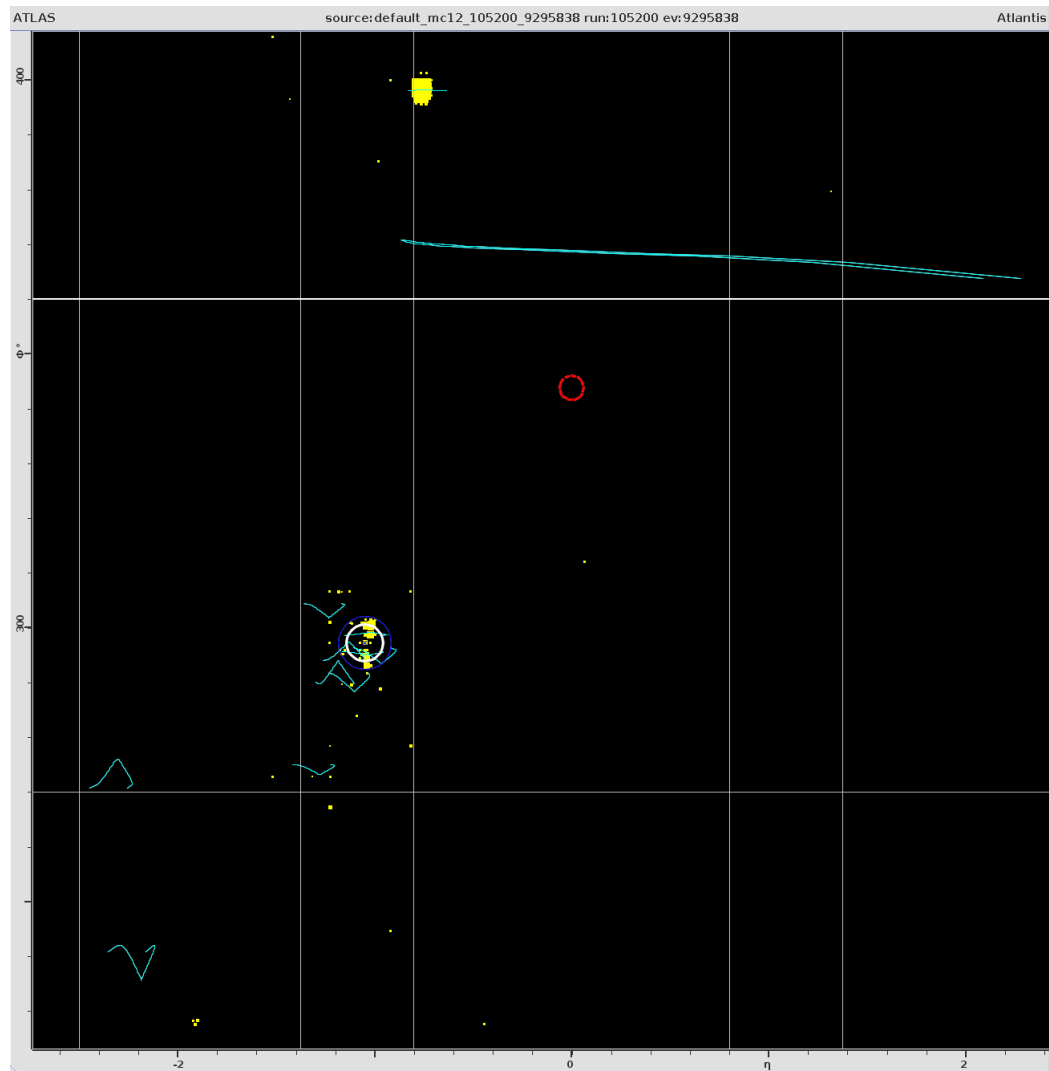
➤ 3D information



For tracks can judge

- ϕ
- η
- pt (slope of V arms)
- charge (Λ -ve
V +ve)

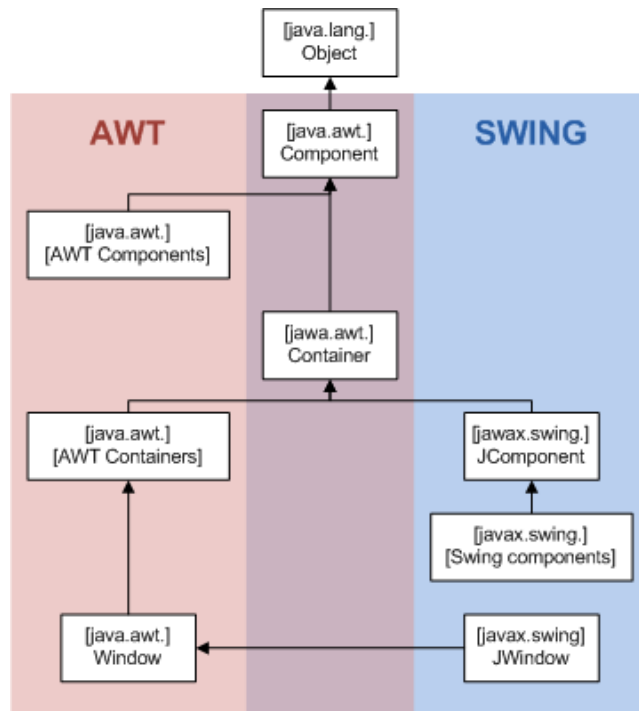
The "V-Plot" in Atlantis



Programming language: Java

- Independent of ATLAS offline framework
- "Run anywhere"
- Similarity to C++
- Web start
- Drawbacks:
 - security concerns
 - decreasing levels of support

Graphics libraries: Swing



- Part of Java Foundation Classes: available wherever Java (SE) is available.

Deployment

- Webstart
 - JNLP: Java Network Launch Protocol
 - User can start application by clicking link in browser
 - Some issues around security, code signing
- Download
 - Can download and run "by hand": `java -jar atlantis.jar`
- CMT
 - Available as part of ATLAS offline software framework
 - Just type `atlantis` once the framework is installed

The Atlantis development team

- Level of support has varied over time: funding, politics, interest of students.
- Currently about 0.5 FTE (full-time equivalent)
 - fraction of time of each developer
 - new developers start, old ones leave
 - varying levels of expertise, both in software and physics
- Led by UCL, with significant contributions from Birmingham and sometimes Nijmegen, CERN, UC Santa Cruz.

The future

- Might depend on what happens in the HEP Software Foundation and wider community.
- Will require development work to maintain functionality
 - whether in Atlantis Java application or elsewhere
 - many other "run anywhere" environments now: browser-based, Unreal Engine...
- Priorities
 - improve handling of associations (e.g. tracks-jets)
 - abstract data format from rest of code: XML has drawbacks
 - refactoring and testing to enable more frequent, reliable releases
- Maybe...
 - separate ATLAS-specific code from generic framework
 - generic framework could be shared more widely...
 - ...or could it?
 - unclear how to deal with licensing of existing collaboration-internal code with many contributors

Conclusions

- There is a place for sophisticated 2D event displays.
- Maintenance is needed, but hard to get developer time funded.
- Could do with guidance on sharing existing code with no explicit licence or contributor agreement.