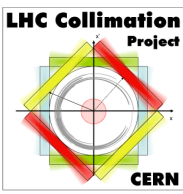


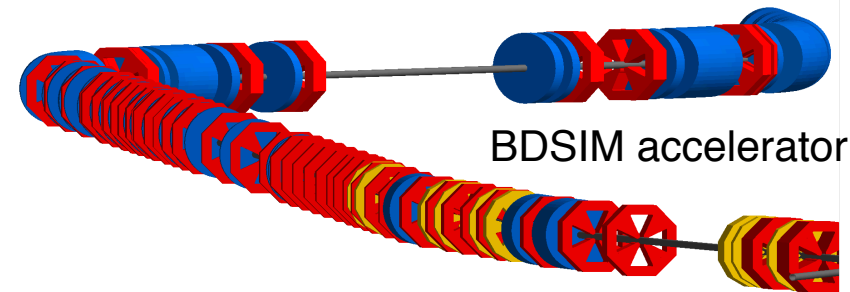
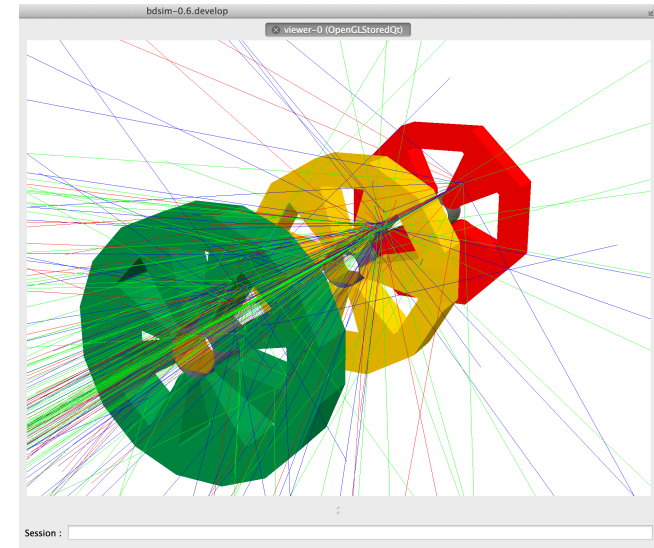
# BDSIM Development 2014 - 2015

**L. Nevay**, J. Snuverink, S. Boogert,  
H. Garcia-Morales, S. Gibson, L. Deacon  
R. Kwee-Hinzmann, S. Walker, A. Abramov

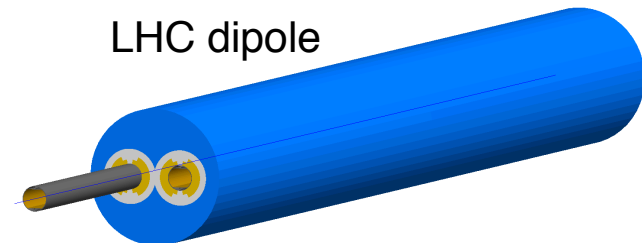


# Beam Delivery Simulation - BDSIM

- Tracking code that uses Geant4
- Open source C++
- Automatically builds Geant4 model
- Uses MadX-like syntax for test input
- Mixes normal accelerator tracking & Monte Carlo particle physics
- Full showers of secondaries created by Geant4 processes
- Ability to simulate synchrotron radiation
- Simulate energy deposition and detector backgrounds
- Ability to import external geometry and field maps

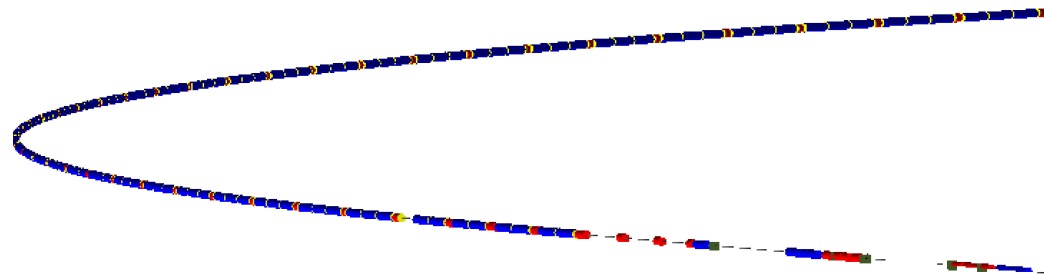
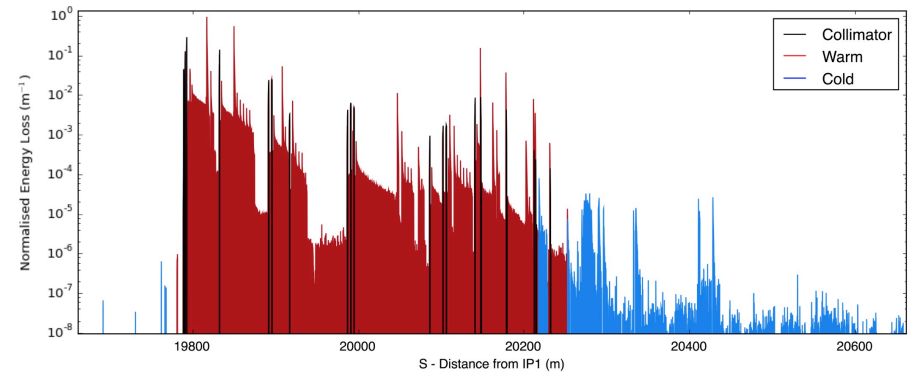
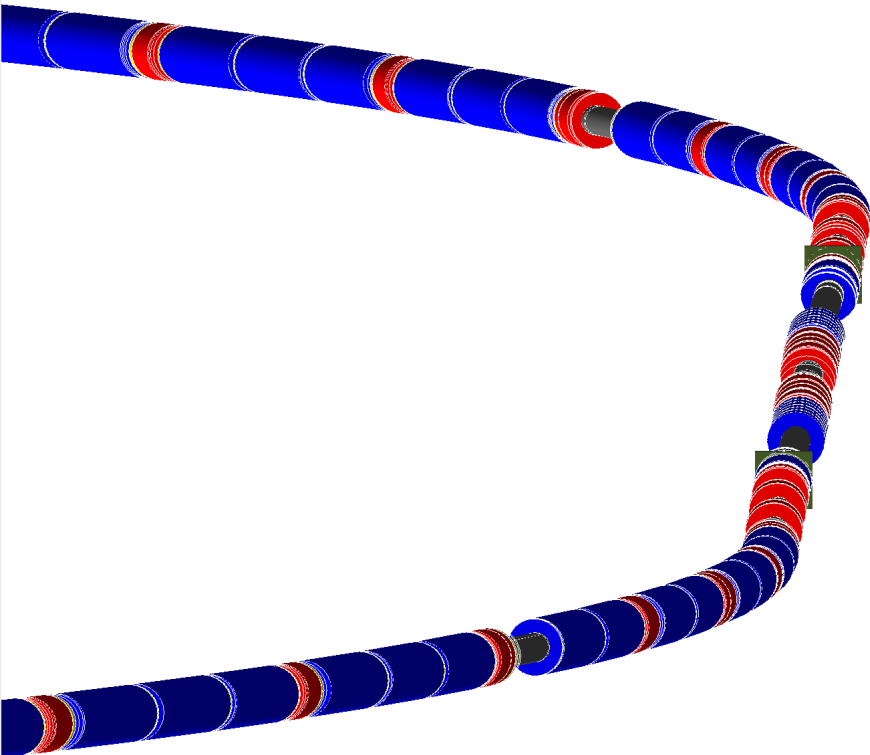


LHC dipole



# Previously...

- Initial conversion and loss maps using BDISM
- First Geant4 based loss map of LHC
- Simplistic geometry
  - symmetric geometry
  - only circular & elliptical aperture



# BDSIM Development

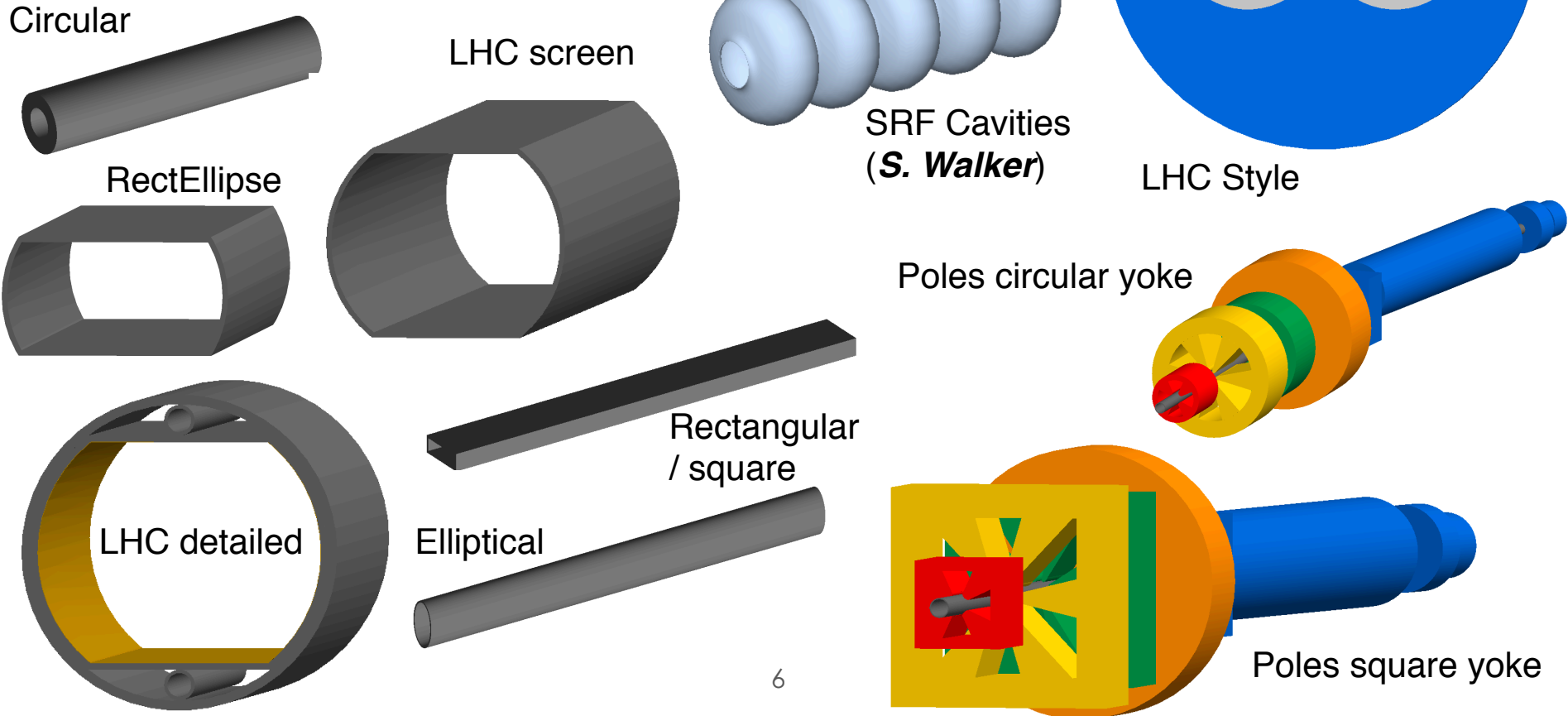
- BDSIM heavily developed since 2012 / 2013
- Recent development followed 3 main themes:
  - ***Geometry*** ✓
  - ***Tracking*** ✓
  - ***Physics processes*** ✓
- Previous questions & issues fell into these three categories
- LHC Specific developments
- Documentation & general development
- Analysis tools & workflow

# Geometry

- Previous geometry relatively simple
  - Adequate for conceptual studies
  - Great detail required for real machines
- Main geometry library rewritten
- Extensive use of factory pattern
  - Each factory represents a style and can make every type of say magnet
- 6 different **aperture** types (including detailed LHC)
- 6 different **magnet styles** (again with LHC style)
- 4 different **tunnel styles**
  - can follow the beam line
  - will be able to have external geometry and customise for certain ranges
- Most importantly all geometry works together
- Any beam pipe will work with any magnet!
- Very simple to extend with new geometry
  - guaranteed to work with all magnets
- New collimators by **H. Garcia-Morales**

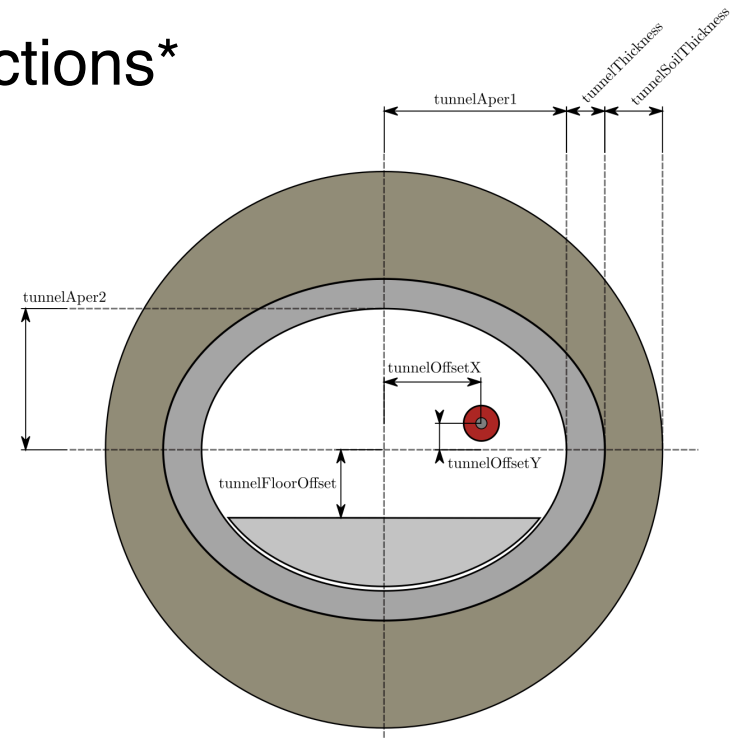
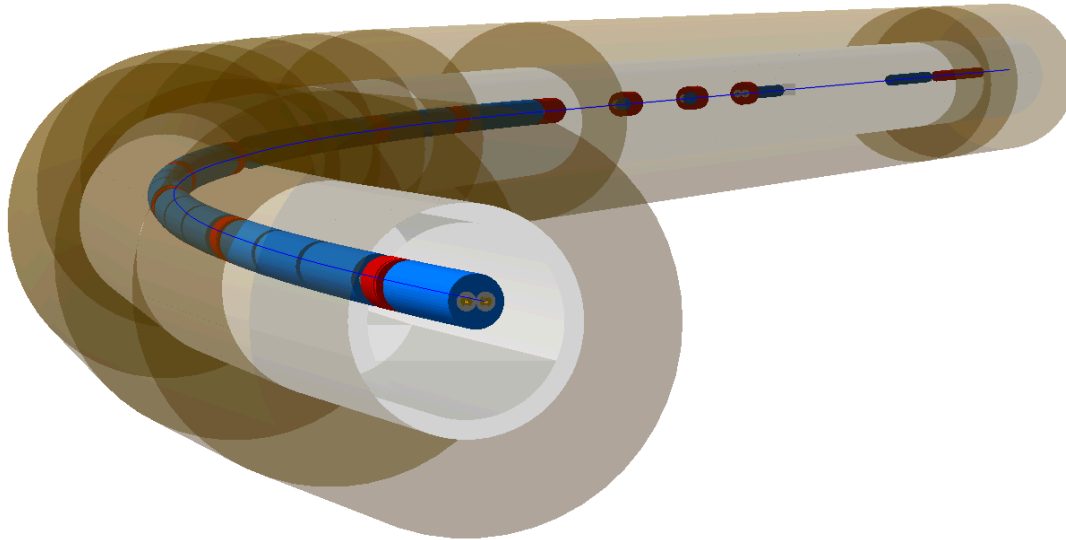
# Geometry

- 6 aperture models
  - circular, rectangular, elliptical, lhc, lhcdetailed, rectellipse
- Modelled on MadX aperture parameterisation
- Works with any other geometry
- 6 different magnet styles



# Tunnel Geometry

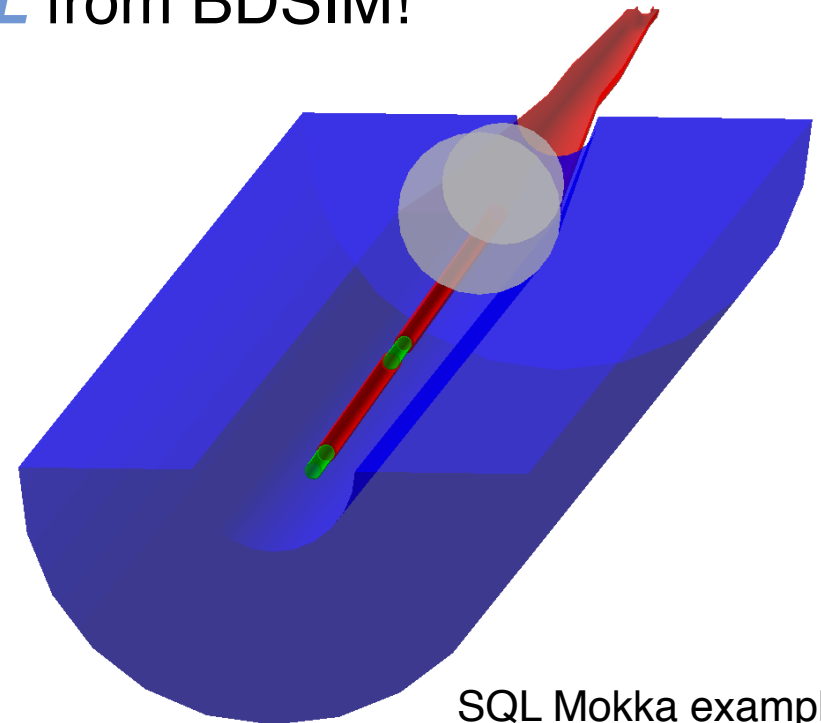
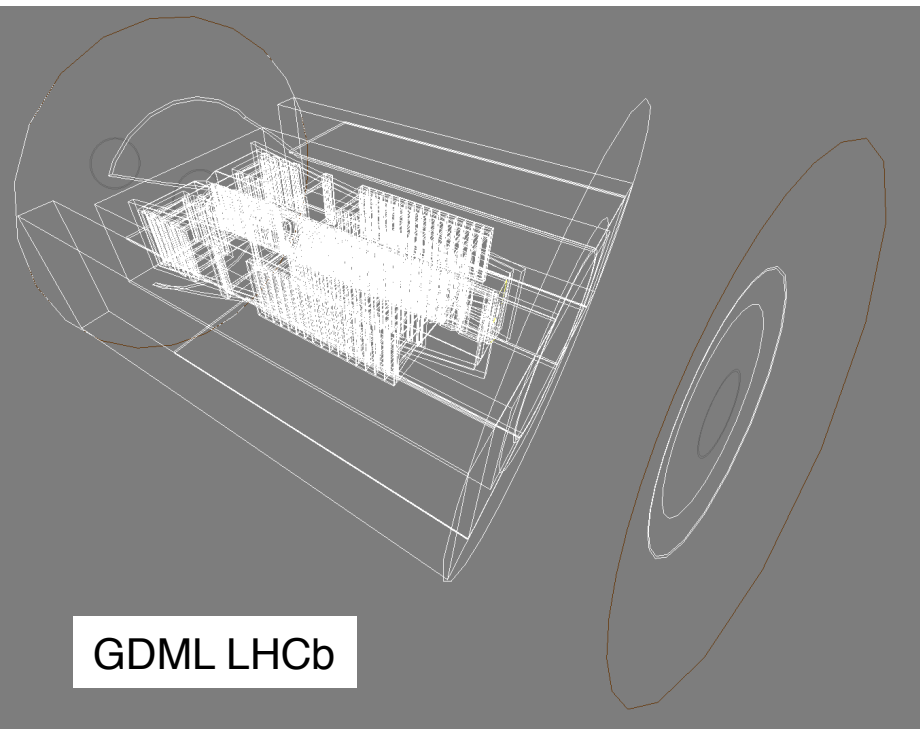
- Was only partially implemented previously
- Rewritten using factories
- Currently 4 different styles
- Can automatically follow beam line
- Can describe different styles for different sections\*
- Can use external geometry for sections\*



\* nearly complete

# External Geometry?

- For when the generic components just won't suffice
- Can import external geometry
  - SQL, Mokka, GDML, STL
- Can also overlay field maps and interpolate
  - 2D, 3D, etc.
- You can also *export to GDML* from BDSIM!

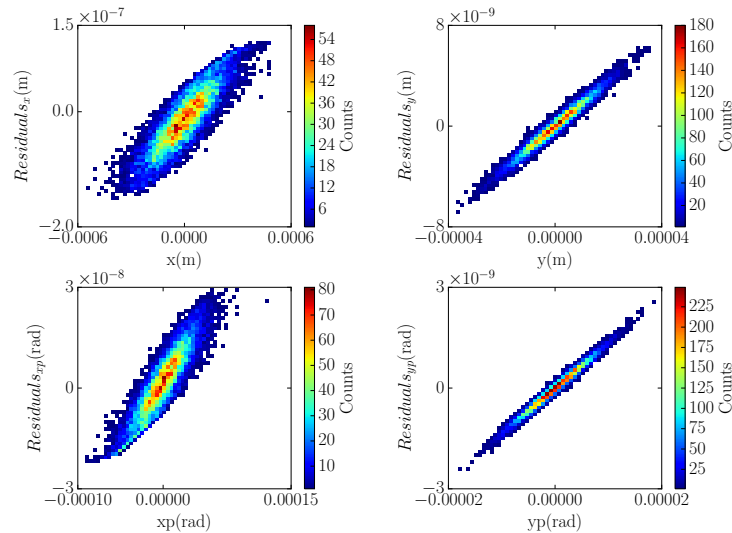




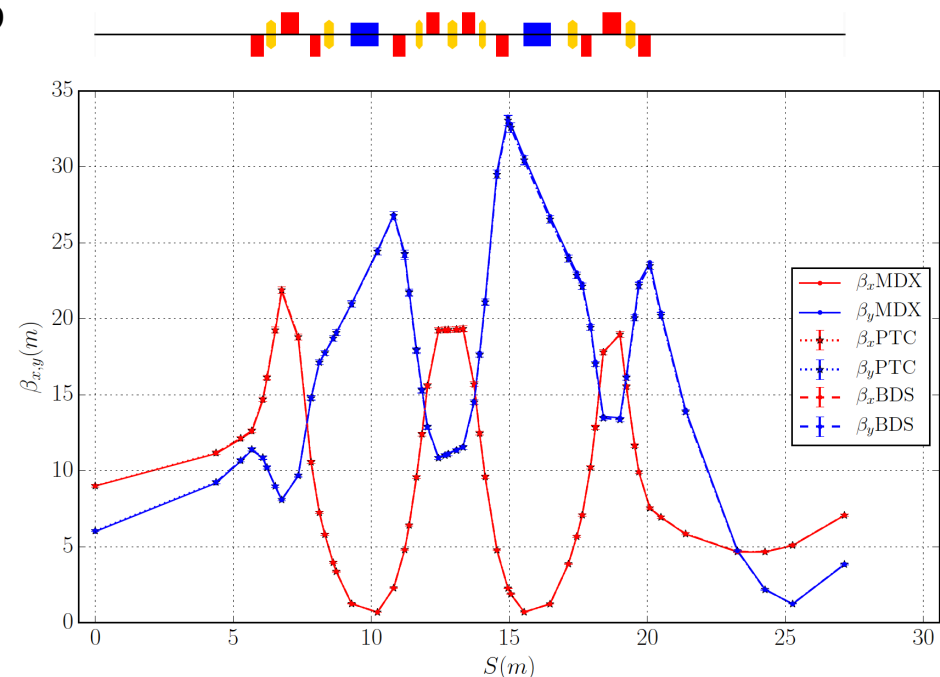
# Tracking

- Quantitative comparison with PTC & SixTrack underway
- Very good agreement with PTC
  - Tracking & optical function calculation
- Factorising tracking into library
  - will reduce tracking time by order of magnitude
  - Will allow choice of integrators
  - Will be able to use other tracking lib
  - Expected complete early 2016

**A. Abramov**



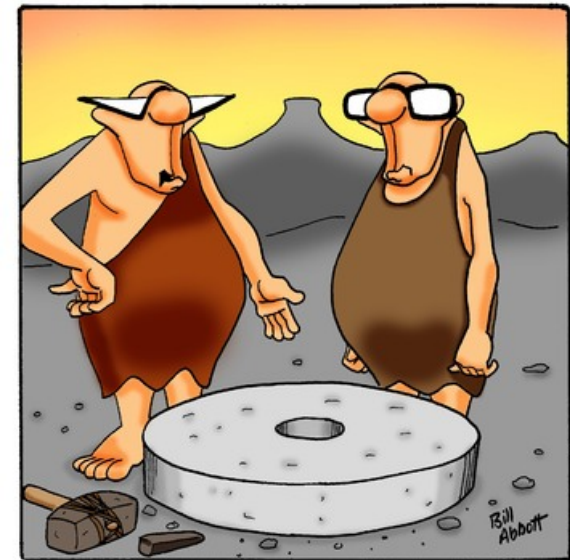
Double Bend Achromat agreement with PTC



# Symplecticity

- Clearly not symplectic
- Factorising into a tracking library opens door to this
- Track first and test aperture continuously
- When approaches aperture -> shift to Geant4 model
- Non-symplectic tracking sufficient from then on
  - impact inherently non-symplectic!
- Faster
  - get the particle to the impact point
- Ideally use an external library!

Current LHC model very stable to 1000s of turns



“I thought I was on to something  
but I can’t figure out how to  
move it.”

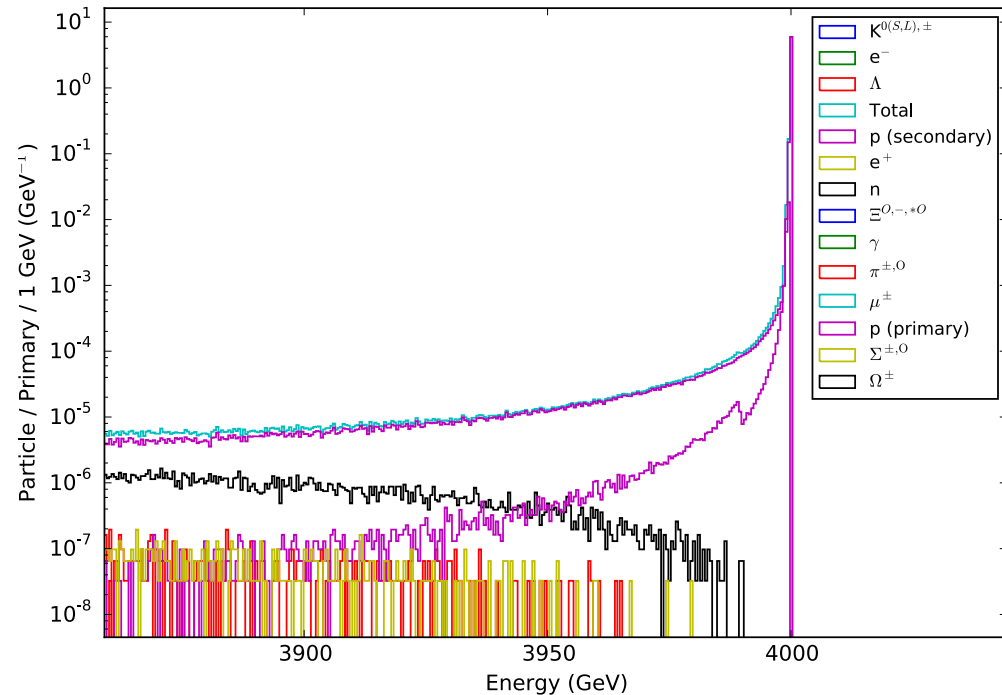
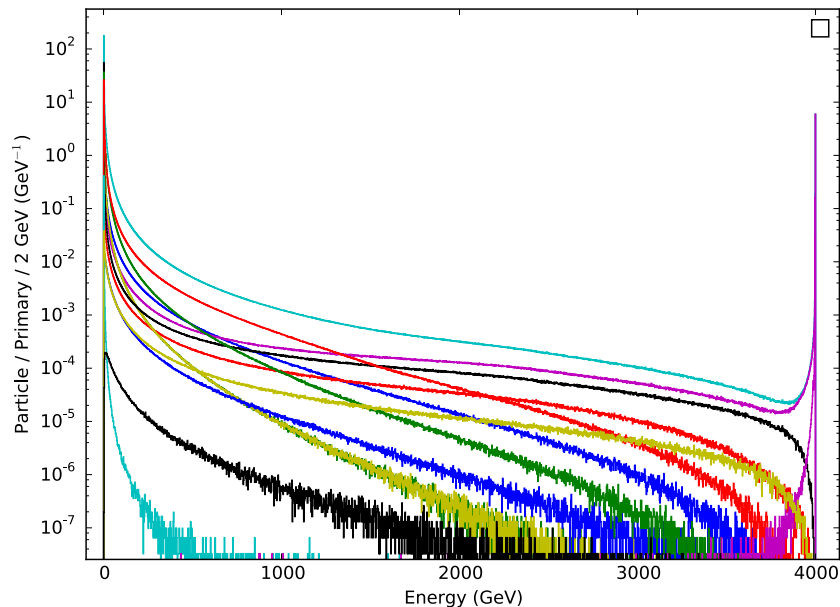
# Physics & Processes



- Benefit from regular Geant4 updates to many many models
- Moved entirely to range cuts
  - several bugs fixed in this
- Cut particles not on energy but on range to produce a secondary particle
- Much more accurate stopping location
  - and therefore energy deposition
- Improved physics accuracy for lower CPU usage
- Modular physics list implemented in Geant4
  - can mix and add to physics processes very easily
- Remember, if it can be wrapped in C++, you can add the physical processes

# Processes

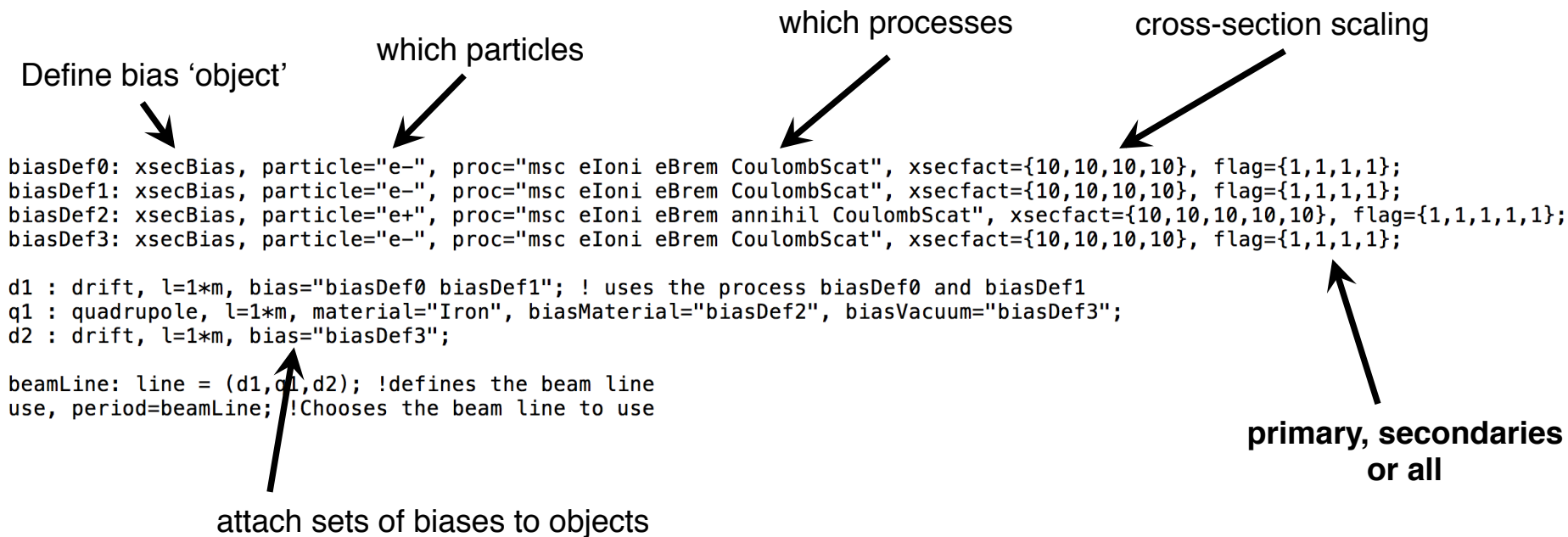
- Quantitative comparison of scattering cross-sections on-going
- Benefiting from updated Geant4 all the time
- Example spectra from scattering at end of TCP
  - $1 \times 10^8$  primaries simulated



# Process Biasing

*S. Boogert*

- Introduced interface to Geant4 process biasing
- **Any** process for **any** particle can be biased for **any** volume or set of volumes



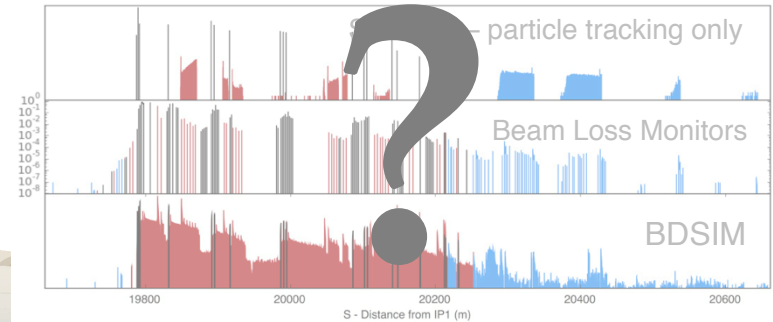
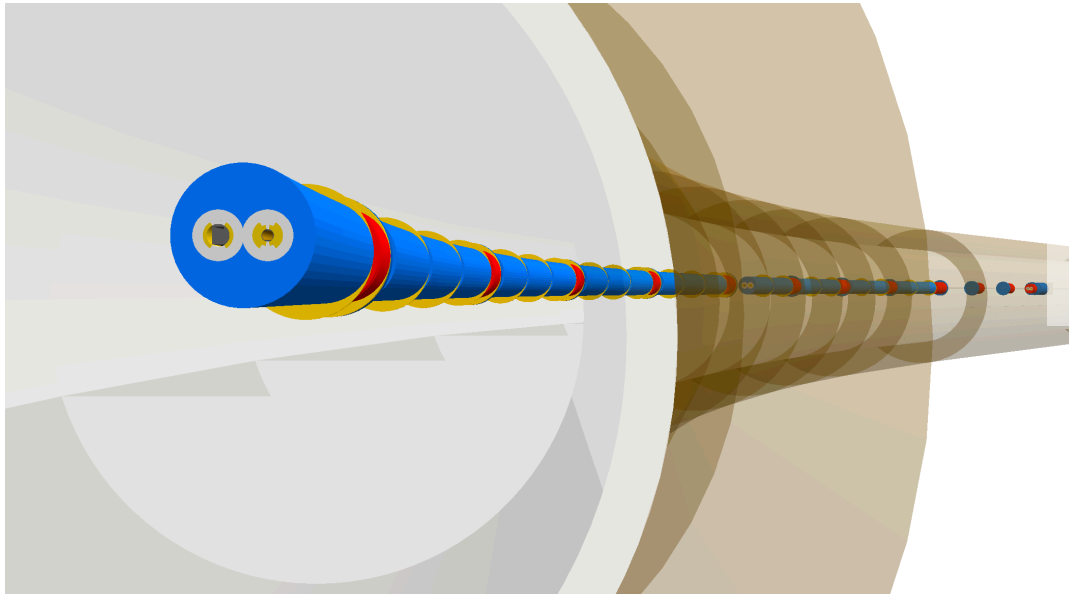
- Extremely flexible interface
- Attach to vacuum or general accelerator material
- Previously required specially written wrapper class for each

# LHC Specific Developments

- Beam pipe geometry & magnet style
- Most development *can* and *has been* very general
- Rewriting of geometry was necessary
  - gives the required aperture flexibility
  - key to correct results!
- Python utilities extend for large automatic conversion
  - definitely required for the LHC
  - now integral part of work flow for all studies
- Primary trouble in piecing together information
- Aperture information significantly different from magnetic
- Tools nearly finished to split and match elements

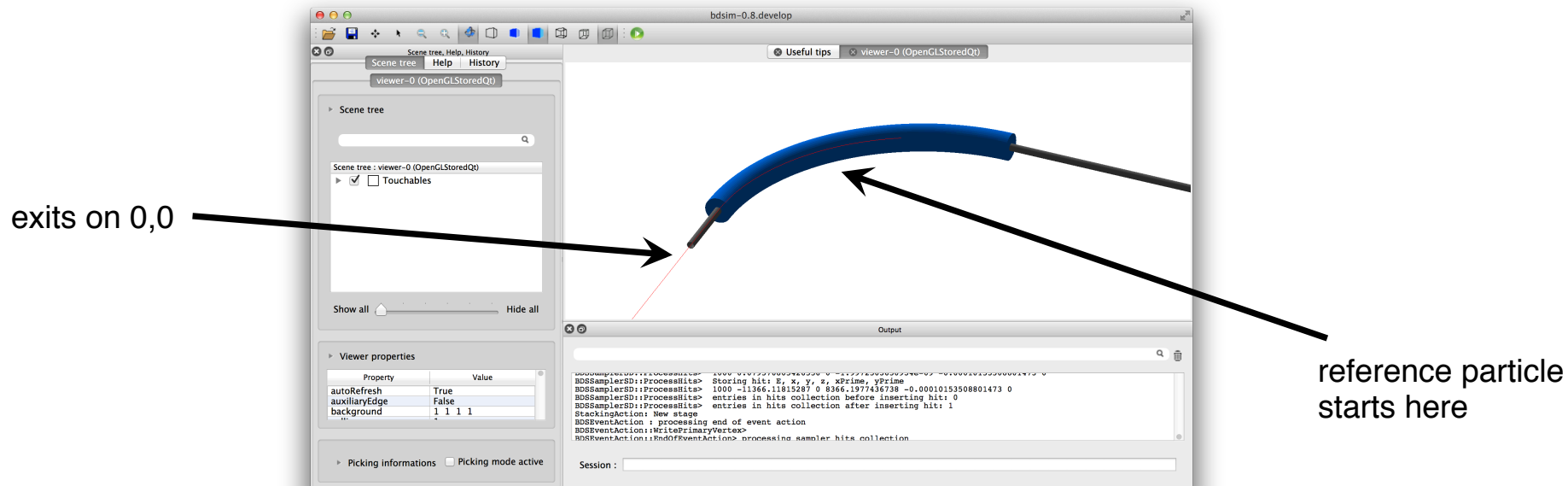
# Loss Map & 6T Hits -> Loss Map

- Unfortunately, this will have to follow
- Significant time to get aperture correct
- Large combination of files in many formats to be converted
- Model must be carefully checked
- Will present at ColUSM in November



# Direct Injection

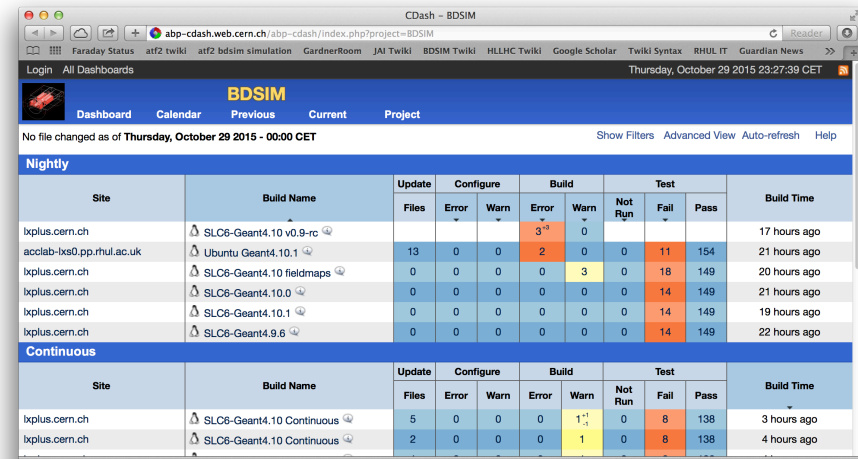
- Had the ability to read out in curvilinear coordinates – now in too
- Introduced ability to inject particles anywhere in lattice
- Any beam distribution as function of  $S$ 
  - Interpolation of trajectory within arcs
  - Efficient look up of transforms
- Sixtrack loader written by ***R. Kwee***
- Can therefore convert SixTrack hits to energy deposition





# Code & Software Development

- 60 000 lines of C++
- Revised *class hierarchy* & *factory patterns* – improved geometry
- Increased factorisation – much easier to extend
- Consolidation of development branches
- C++ 11 adoption & latest versions of Geant4, ROOT, CLHEP,
- Parser significantly revised by **J. Snuverink**
  - memory leaks, and problems fixed
  - written in object-orientated C++
- CTest test suite, CMake build system
  - much easier to use as compared to old configuration scripts
- CDash **nightly** and **on demand** automated building & testing
- Issue tracking & reporting
- Built in configuration for AFS
- Automated manual updates
- Regular release cycle



CDash - BDSIM

abp-cdash.web.cern.ch/abp-cdash/index.php?project=BDSIM

Dashboard Calendar Previous Current Project

No file changed as of Thursday, October 29 2015 - 00:00 CET

Site Build Name Update Configure Build Test Build Time

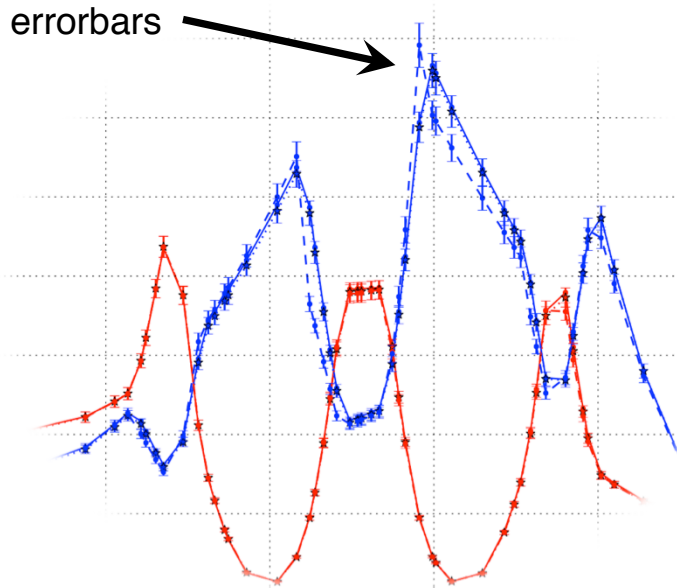
Site	Build Name	Update		Configure			Build			Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass				
ixplus.cern.ch	SLC6-Geant4.10 v0.9-rc				3	0						17 hours ago	
acclab-ixs0.pp.rhul.ac.uk	Ubuntu-Geant4.10.1	13	0	0	2	0	0	0	11	154		21 hours ago	
ixplus.cern.ch	SLC6-Geant4.10 fieldmaps	0	0	0	0	3	0	0	18	149		20 hours ago	
ixplus.cern.ch	SLC6-Geant4.10.0	0	0	0	0	0	0	0	14	149		21 hours ago	
ixplus.cern.ch	SLC6-Geant4.10.1	0	0	0	0	0	0	0	14	149		19 hours ago	
ixplus.cern.ch	SLC6-Geant4.9.6	0	0	0	0	0	0	0	14	149		22 hours ago	
Continuous													
Site	Build Name	Update		Configure			Build			Test			Build Time
ixplus.cern.ch	SLC6-Geant4.10 Continuous	5	0	0	0	1	0	0	8	138		3 hours ago	
ixplus.cern.ch	SLC6-Geant4.10 Continuous	2	0	0	0	1	0	0	8	138		4 hours ago	

# Output Analysis

- ASCII & ROOT output formats
- ASCII useful for initial and single particle trials
  - ASCII isn't suited to large data
  - ASCII inefficient storage and not as strongly structured
- ROOT output used for studies
  - ability to introduce other formats if / when required (HDF5)
- RoBDSIM analysis tool written in C++
  - compiled C++
- Can use in 3 ways with exactly the same functionality
  - C++ linked to, use interactively in ROOT, use in python
- RoBDSIM typically used in Python, ROOT and on farm
- Can add and produce histograms
- Suitable for farm analysis of job files
- <https://bitbucket.org/jairhul/robdsim>

# Statistical Comparison of Tracking

- Can calculate optical functions from particle distribution
- Another measure as compared to single particle tests
- Used to compare BDSIM with design of lattice
  - useful for spotting conversion discrepancies
- Can compare with PTC
- Significant work on uncertainty calculation of optical functions from particle distribution



$$\epsilon_x = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle + \langle xx' \rangle^2}, \quad \Sigma = \begin{pmatrix} \langle x^2 \rangle & \langle xx' \rangle \\ \langle x'x \rangle & \langle x'^2 \rangle \end{pmatrix}$$

$$\beta_x = \frac{\langle x^2 \rangle}{\epsilon_x}$$

$$\sigma_{\vec{f}}^2 = \left( \frac{\partial \vec{f}}{\partial a_1} \right)^2 cov[a_1, a_1] + \left( \frac{\partial \vec{f}}{\partial a_2} \right)^2 cov[a_2, a_2] + \left( \frac{\partial \vec{f}}{\partial a_3} \right)^2 cov[a_3, a_3] +$$

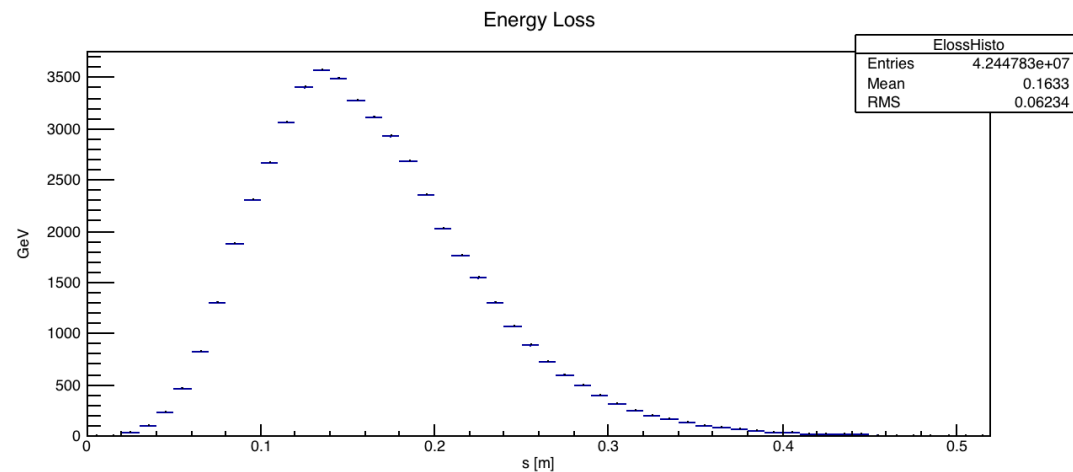
$$+ 2 \left( \frac{\partial \vec{f}}{\partial a_1} \frac{\partial \vec{f}}{\partial a_2} \right) cov[a_1, a_2] + 2 \left( \frac{\partial \vec{f}}{\partial a_1} \frac{\partial \vec{f}}{\partial a_3} \right) cov[a_1, a_3] + 2 \left( \frac{\partial \vec{f}}{\partial a_2} \frac{\partial \vec{f}}{\partial a_3} \right) cov[a_2, a_3].$$

$$cov[a_3, a_3] = -\frac{(-2+n)\mu_{1,1}^2}{(-1+n)n} + \frac{\mu_{0,2}\mu_{2,0}}{(-1+n)n} + \frac{\mu_{2,2}}{n},$$

# Regression Testing

- Rapid development of BDSIM
- Occasionally, simple / basic things break
- Code too large to test all features yourself
- Automated build & testing system implemented
- Each example is also a test
- Reference histograms and results compared
- RoBDSIM executable used for comparison and testing
- **145 tests** so far
- **Run nightly**

- Hadronic & EM Shower development
- Tracking in each component
- Parser
- Geometry construction
- Geometry overlaps
- Many more....



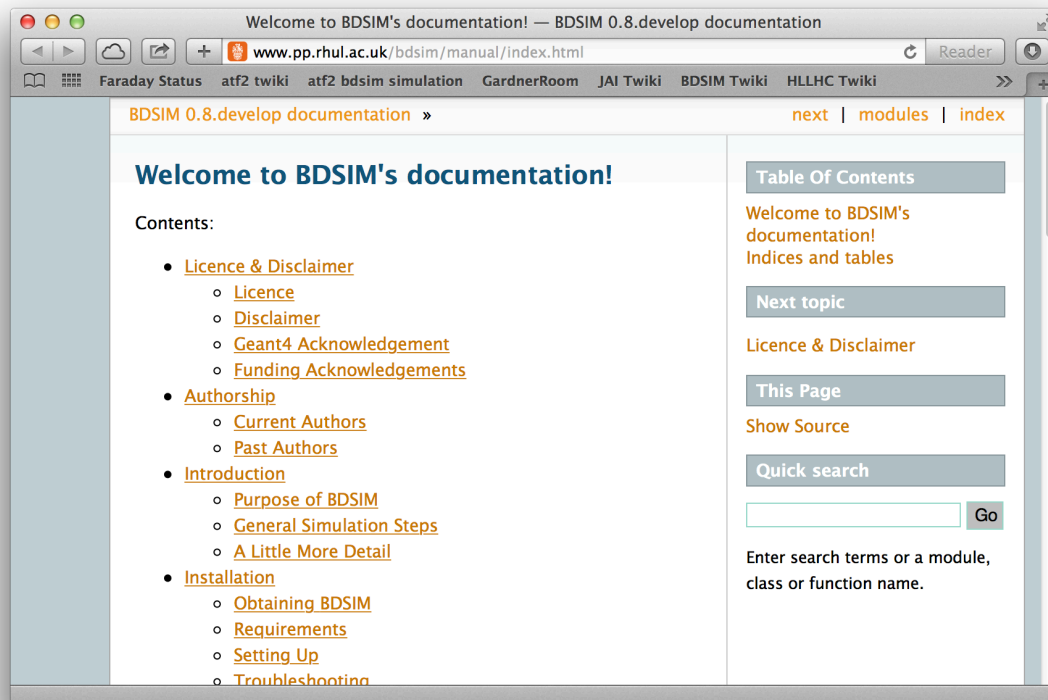
# Python Utilities Galore



- pymadx, pybdsim, pymad8, robdsim, pytransport, pylhc
- pymadx
  - loading and manipulation of TFS files
  - range iterating, filtering, matching
  - PTC segments supported
  - use to plot a lattice above a graph – interactive too!
- pybdsim
  - conversion from Madx, Mad8, Transport etc
  - ASCII output analysis
  - programmatic model construction
- pylhc
  - utilities for parsing Lhc model specific information
  - collimation files, aperture information (filtering, matching etc)
- Again, all open source and distributed with BDSIM

# Documentation

- New manual (html & pdf) automatically updated weekly
  - lots of syntax examples
  - [www.pp.rhul.ac.uk/bdsim/manual](http://www.pp.rhul.ac.uk/bdsim/manual)
- Detailed Doxygen code documentation similarly
  - [www.pp.rhul.ac.uk/bdsim/doxygen](http://www.pp.rhul.ac.uk/bdsim/doxygen)



Welcome to BDSIM's documentation! — BDSIM 0.8.develop documentation

[www.pp.rhul.ac.uk/bdsim/manual/index.html](http://www.pp.rhul.ac.uk/bdsim/manual/index.html)

Faraday Status atf2 twiki atf2 bdsim simulation GardnerRoom JAI Twiki BDSIM Twiki HLLHC Twiki

BDSIM 0.8.develop documentation » next | modules | index

## Welcome to BDSIM's documentation!

Contents:

- [Licence & Disclaimer](#)
  - [Licence](#)
  - [Disclaimer](#)
  - [Geant4 Acknowledgement](#)
  - [Funding Acknowledgements](#)
- [Authorship](#)
  - [Current Authors](#)
  - [Past Authors](#)
- [Introduction](#)
  - [Purpose of BDSIM](#)
  - [General Simulation Steps](#)
  - [A Little More Detail](#)
- [Installation](#)
  - [Obtaining BDSIM](#)
  - [Requirements](#)
  - [Setting Up](#)
  - [Troubleshooting](#)

Table Of Contents

Welcome to BDSIM's documentation!  
Indices and tables

Next topic

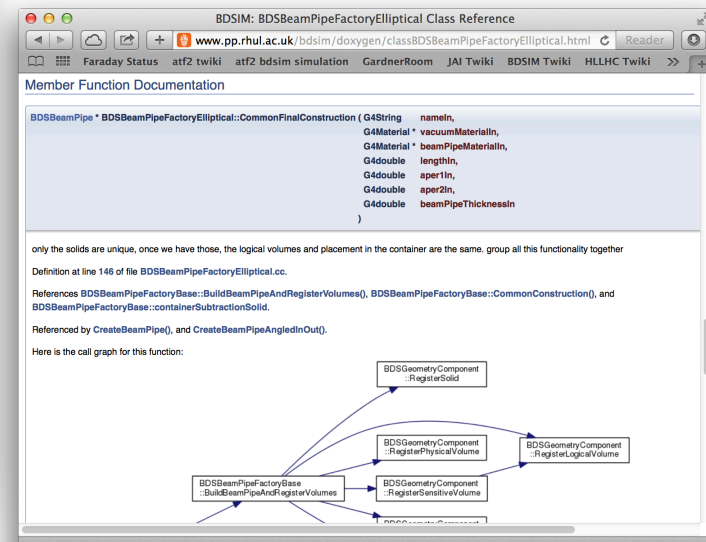
Licence & Disclaimer

This Page

Show Source

Quick search

Enter search terms or a module, class or function name.



BDSIM: BDSBeamPipeFactoryElliptical Class Reference

[www.pp.rhul.ac.uk/bdsim/doxygen/classBDSBeamPipeFactoryElliptical.html](http://www.pp.rhul.ac.uk/bdsim/doxygen/classBDSBeamPipeFactoryElliptical.html)

Faraday Status atf2 twiki atf2 bdsim simulation GardnerRoom JAI Twiki BDSIM Twiki HLLHC Twiki

### Member Function Documentation

```
BDSBeamPipe * BDSBeamPipeFactoryElliptical::CommonFinalConstruction ( G4String nameIn,
                                                                    G4Material * vacuumMaterialIn,
                                                                    G4Material * beamPipeMaterialIn,
                                                                    G4double lengthIn,
                                                                    G4double aperIn,
                                                                    G4double aperZIn,
                                                                    G4double beamPipeThicknessIn
                                                                    )
```

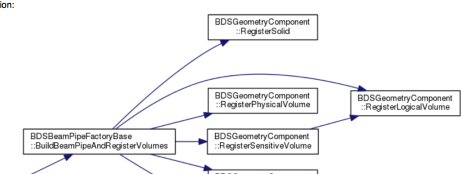
only the solids are unique, once we have those, the logical volumes and placement in the container are the same. group all this functionality together

Definition at line 146 of file BDSBeamPipeFactoryElliptical.cc.

References BDSBeamPipeFactoryBase::BuildBeamPipeAndRegisterVolumes(), BDSBeamPipeFactoryBase::CommonConstruction(), and BDSBeamPipeFactoryBase::containerSubtractionSolid.

Referenced by CreateBeamPipe(), and CreateBeamPipeAngledInOut().

Here is the call graph for this function:



```
graph TD
    A["BDSBeamPipeFactoryBase  
:BuildBeamPipeAndRegisterVolumes"] --> B["BDSGeometryComponent  
:RegisterSolid"]
    A --> C["BDSGeometryComponent  
:RegisterPhysicalVolume"]
    A --> D["BDSGeometryComponent  
:RegisterSensitiveVolume"]
    A --> E["BDSGeometryComponent  
:RegisterLogicalVolume"]
```

# Public Git Access

- [www.bitbucket.org/jairhul/bdsim](http://www.bitbucket.org/jairhul/bdsim)
- Full open source development
- Issue tracking - (100 this year, 20 open)
- ~ 10 regular developers
- ~ 5 branches

300 – 500 commits per version  
3 releases per year typically

```

1941641 rename BDSMagnetColours to BDSColours
4602763 fix comment
9052aee remove unused file
f225beb unused variable warning fix - left over from merge with develop
c382925 compiler fixes from merging
3d09fe8 move new examples to avoid clash with existing naming scheme in develop
bc1f6dd Merge branch 'develop' into fieldmaps

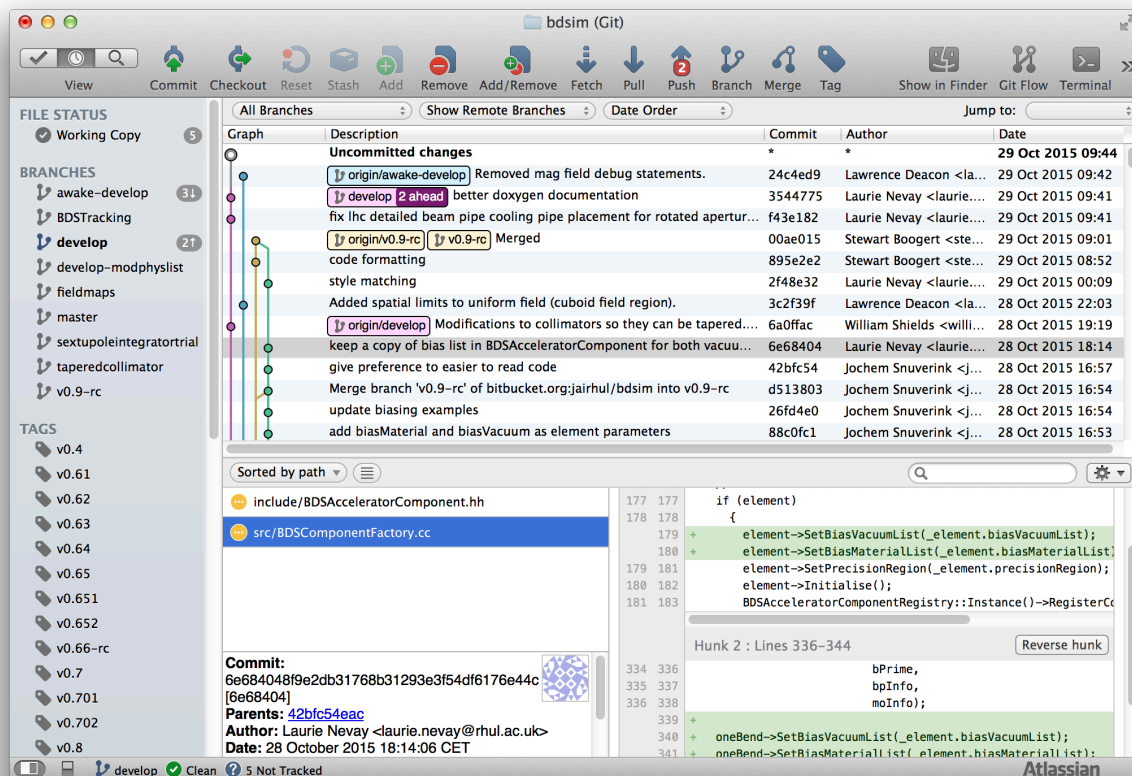
4de088b increase tolerance for read out volume overlapping faces for strong angles with short length bends. Fixes issue #73.
5fb49dd stricter checking on bending magnet parameters to avoid overlapping entrance and exit faces
29df7fa allow 0 angle sbends with poled geometry - will only do poled and cylindrical - lhc more complicated and always bent. Fixes issue #74. No overlaps found.
4ffde47 allow 0 angle sbends with cylindrical magnet geometry
abac782 avoid sbend geometry warnings for 0 angle and correct length calculation
e9020ba define a sensible size in gdm1 for 'world' volume instead of 100m radius volume that causes overlaps with rest of beam line and containers
176c525 tidying and debug visualisation for gdm1 world volume
16bacdf Merge branch 'develop' of https://bitbucket.org/jairhul/bdsim into develop

358c3b4 Merge branch 'develop' of bitbucket.org:jairhul/bdsim into develop

579affc Added degrader to the manual.
5b7419c Minor degrader maths change and remove cout lines.
077550b fix ROOT output for samplers attached to multiple instances of elements (tree name has to be unique, otherwise it will be overwritten)
6097981 add useful command

99a477e fix degraderOffset in parser
865b196 Degraded diagram for manual.
467224f Merge branch 'develop' of https://bitbucket.org/jairhul/bdsim into develop

2d6d758 cleaning and comments
ddfc80 beam command note; storetrajectories update
bd16a12 change default value of maximum radius size for storing trajectories to a large value
2d5f084 introduce features/options directory; example for storetrajectories
3a02b4d remove duplicate option
54d26c7 Change to degrader to take a horizontal offset as an input parameter.
f5eb0bb allow sbend debugging more easily by controlling in input file whether to split bends o
  
```



Graph	Description	Commit	Author	Date
	<b>Uncommitted changes</b>	*	*	29 Oct 2015 09:44
	origin/awake-develop Removed mag field debug statements.	24c4ed9	Lawrence Deacon <la...	29 Oct 2015 09:42
	develop [2 ahead] better doxygen documentation	3544775	Laurie Nevay <laurie...	29 Oct 2015 09:41
	fix lhc detailed beam pipe cooling pipe placement for rotated aper...	f43e182	Laurie Nevay <laurie...	29 Oct 2015 09:41
	origin/v0.9-rc [v0.9-rc] Merged	00ae015	Stewart Boogert <ste...	29 Oct 2015 09:01
	code formatting	895e2e2	Stewart Boogert <ste...	29 Oct 2015 08:52
	style matching	2f48e32	Laurie Nevay <laurie...	29 Oct 2015 00:09
	Added spatial limits to uniform field (cuboid field region).	3c2f39f	Lawrence Deacon <la...	28 Oct 2015 22:03
	origin/develop Modifications to collimators so they can be tapered....	6a0ffac	William Shields <willi...	28 Oct 2015 19:19
	keep a copy of bias list in BDSAcceleratorComponent for both vacu...	6e68404	Laurie Nevay <laurie...	28 Oct 2015 18:14
	give preference to easier to read code	42bfc54	Jochem Snuerink <j...	28 Oct 2015 16:57
	Merge branch 'v0.9-rc' of bitbucket.org:jairhul/bdsim into v0.9-rc	d513803	Jochem Snuerink <j...	28 Oct 2015 16:54
	update biasing examples	26fd4e0	Jochem Snuerink <j...	28 Oct 2015 16:54
	add biasMaterial and biasVacuum as element parameters	88c0fc1	Jochem Snuerink <j...	28 Oct 2015 16:53

Many developers working at once without issue on many versions

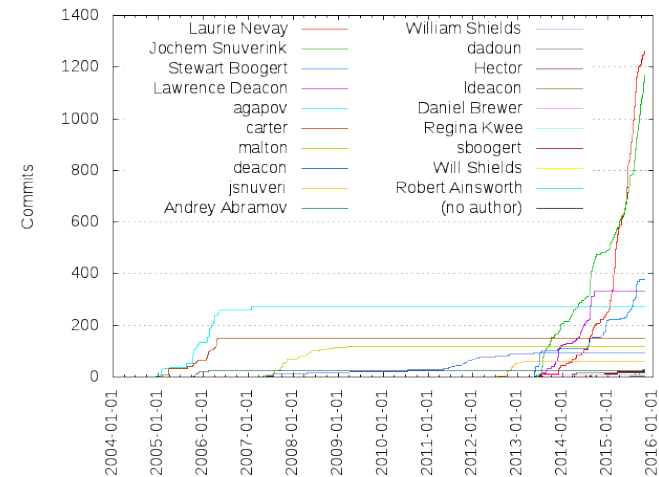
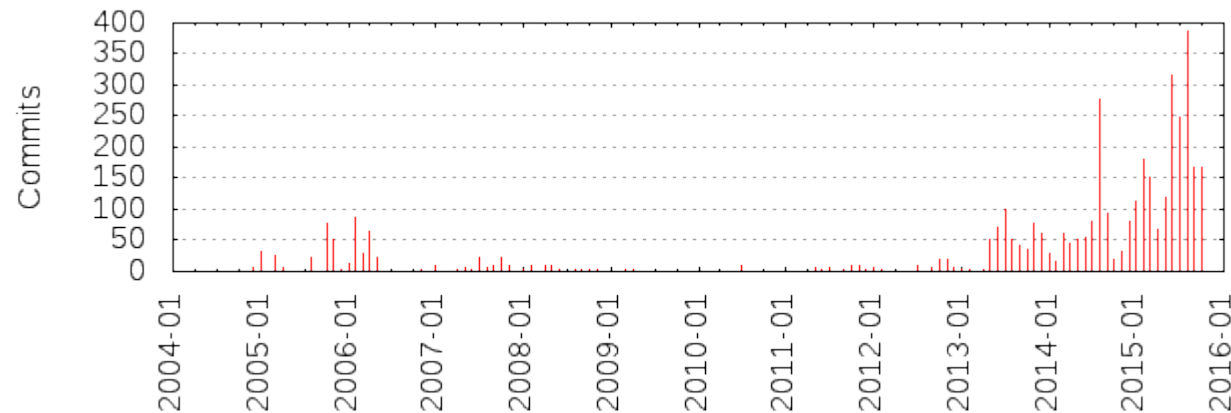
# Timeline

- Autumn 2015
  - v0.9 released *next week*
  - Finish magnet support & BLM geometry
  - Implement magnet coils
  - Implement magnet pole face rotation flexibility
  - Implement Racetrack aperture
  - Complete regression testing suite
  - Support of crystal collimation processes & geometry
- V1.0 by end of 2015
  - paper in preparation
- Spring 2016
  - Factorise tracking
  - Include external tracking library
  - Consider incorporating collimation specific processes



# Summary

- BDSIM has great potential for a variety of applications
- Complimentary to existing codes
- Under rapid development – 7 regular developers
- Flexible and easy to explore new scenarios
- Weekly meetings & bi-annual code weeks
- Clear road map for development
- Paper in preparation



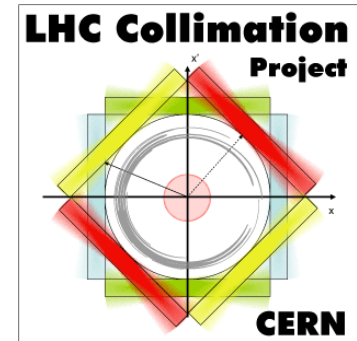


# Thank you

Acknowledgements:  
Regina, Stefano,  
Roderik, Andrea, Hector



[www.pp.rhul.ac.uk/bdsim](http://www.pp.rhul.ac.uk/bdsim)



# Reference Slide for URLs



- ***We've moved!***
- [www.bitbucket.org/stewartboogert/bdsim](http://www.bitbucket.org/stewartboogert/bdsim)
- [www.bitbucket.org/Inevay/pybdsim](http://www.bitbucket.org/Inevay/pybdsim)
  
- [www.pp.rhul.ac.uk/bdsim](http://www.pp.rhul.ac.uk/bdsim)
- [www.pp.rhul.ac.uk/bdsim/manual](http://www.pp.rhul.ac.uk/bdsim/manual)
- [www.pp.rhul.ac.uk/bdsim/doxygen](http://www.pp.rhul.ac.uk/bdsim/doxygen)
- [www.bitbucket.org/jairhul/bdsim](http://www.bitbucket.org/jairhul/bdsim)
- [www.bitbucket.org/jairhul/pymadx](http://www.bitbucket.org/jairhul/pymadx)
- [www.bitbucket.org/jairhul/pybdsim](http://www.bitbucket.org/jairhul/pybdsim)
- <http://abp-cdash.web.cern.ch/abp-cdash/index.php?project=BDSIM>