Beam Delivery Simulation

L. Nevay

laurie.nevay@cern.ch

on behalf of BDSIM group:

A. Abramov, S. Boogert, S. Gibson, H. Garcia-Morales,
H. Pikhartova, W. Shields, J. Snuverink, S. Walker

17th September 2018

Outline

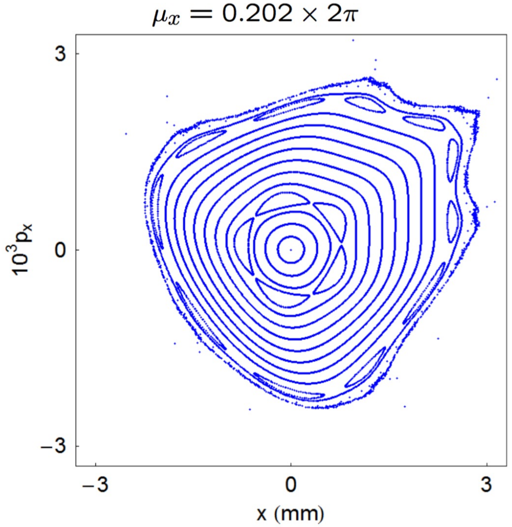
- Brief introduction to the problem
- Description of BDSIM code
- Examples

Introduction

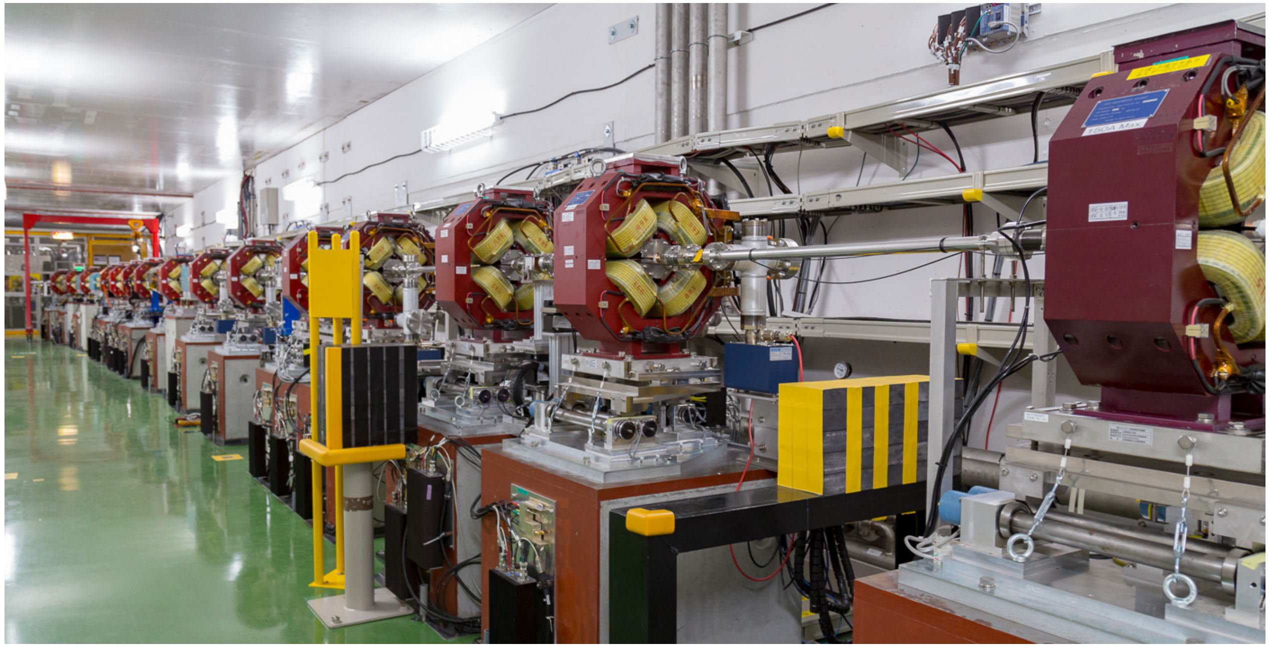
- Beam loss ➔ no accelerator perfectly contains all particles
  - either by design tolerance (certain capture %)
  - via stochastic processes (intra-beam scattering, beam-beam, collective effects)
- Secondary beams
  - beam intentionally hits target to produce secondary short-lived particles
  - beams may be partly transported in air
- Both cases require simulation that includes ***both*** accelerator tracking and interaction with matter
  - accelerator tracking cannot predict interaction with matter or production of secondary particles
  - radiation transport model does provide accelerator tracking and is highly labour intensive to make

Accelerator Tracking

- Electromagnets used to guide particles
  - variety of types, each with different strengths
- For non-uniform B / E field use numerical integration
  - however, slow and limited accuracy
  - not useful for many thousands of operations - error increases
- Specific fields can have specific solutions
- Require physical accuracy and strict energy conservation

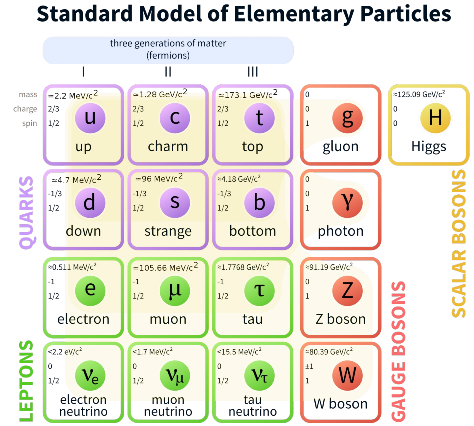


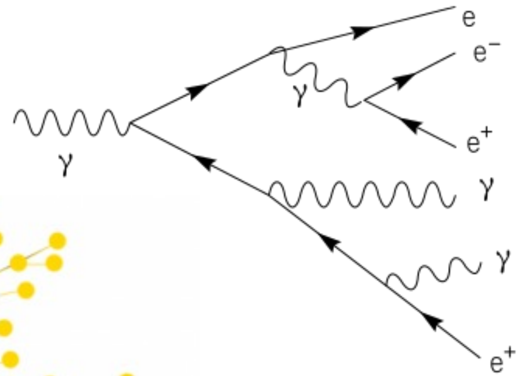
Example nonlinear fields
A. Wolski Lectures at Cockcroft

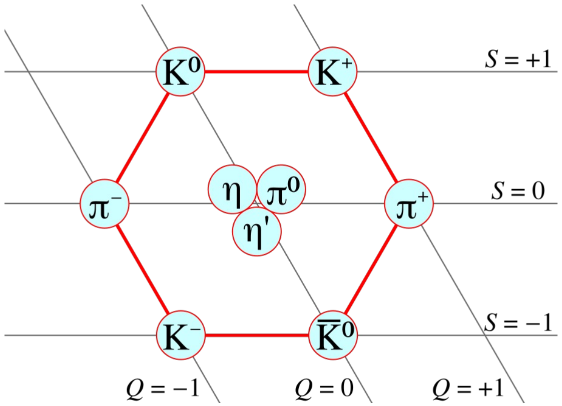


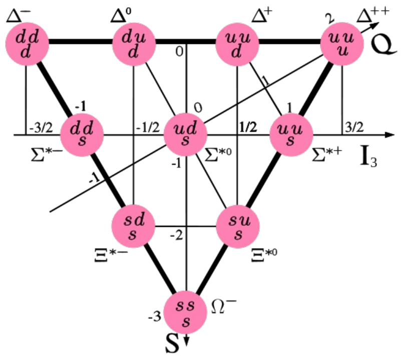
Particle Matter Interaction

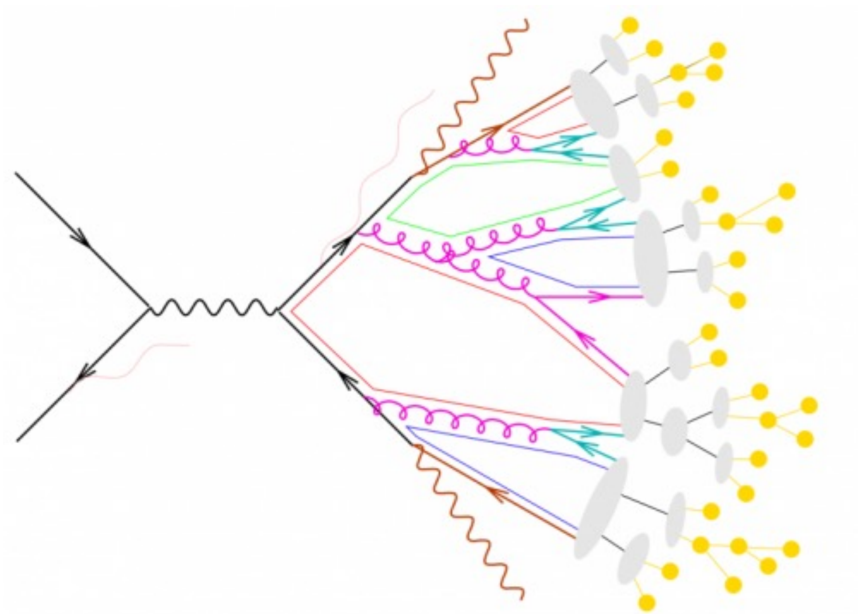
- Large number of subatomic particles
- Large variety of processes & models
  - some data based, some pure model based, some mixed
  - different models for different energy ranges
- Available libraries - FLUKA, Geant4, MARS











Existing Simulation Strategies

- Specialised codes for accelerator tracking or radiation transport models
- Current solutions use a variety of approaches:
  - track up to impact on aperture
  - simulate most relevant parts separately
  - pass between codes

Accelerator Tracking	Radiation Transport
• SixTrack	• FLUKA
• PTC / MADX	• Geant4
• Transport	• MARS
• Lucretia	• MCNPX

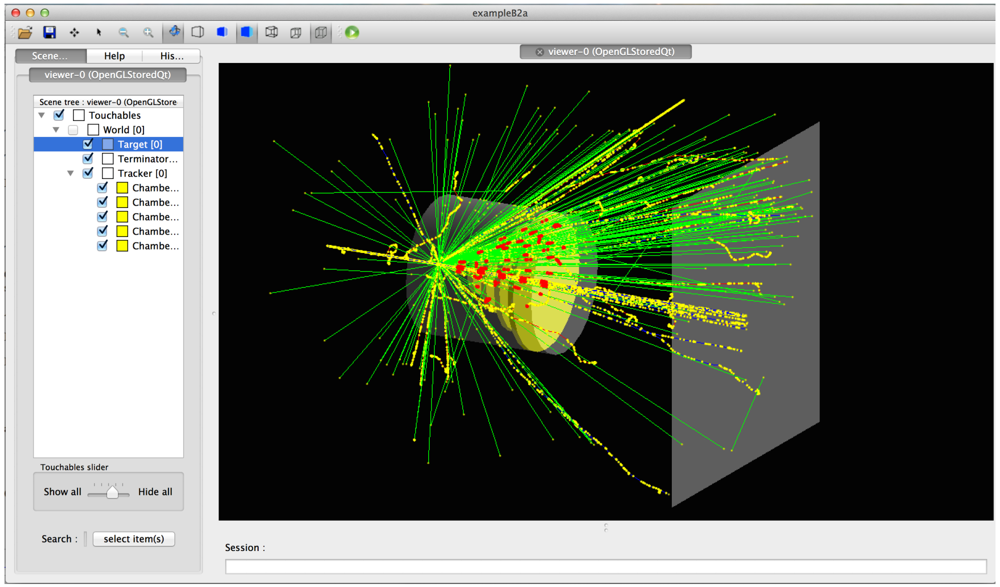
Which Physics Package?

- Geant4
  - open source C++ class library
  - no executable program
  - conceived to simulate particle detector response
  - extensive particle physics models
  - regularly updated ~ every 6 months
  - used by detector community

- FLUKA
  - ASCII input
  - also extensive particle physics models
  - used by radiation shielding community
  - closed source Fortran
  - highly restrictive licence

GEANT4
A SIMULATION TOOLKIT

http://geant4.web.cern.ch



Geant4 example of proton hitting calorimeter

Complexity…

- Creating 3D model of an accelerator is laborious
- Many people many years work
- Hard coded to that application
- Complex to create and validate

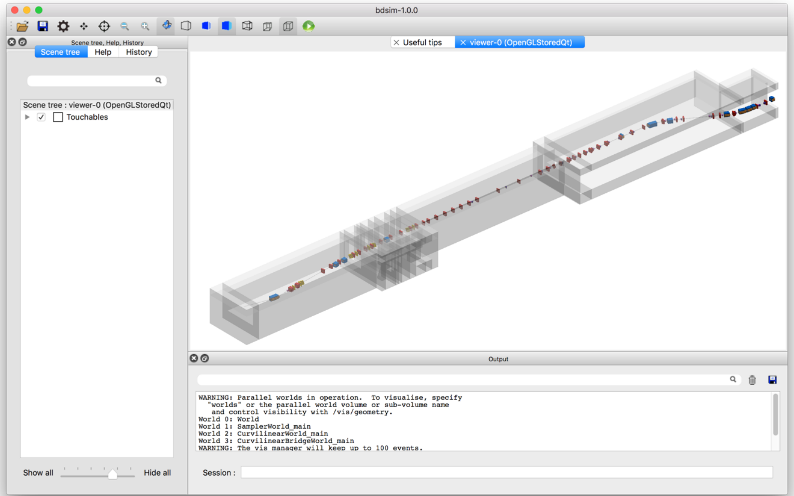
- Tracking codes complex in implementation
- Speciality can vary depending on application

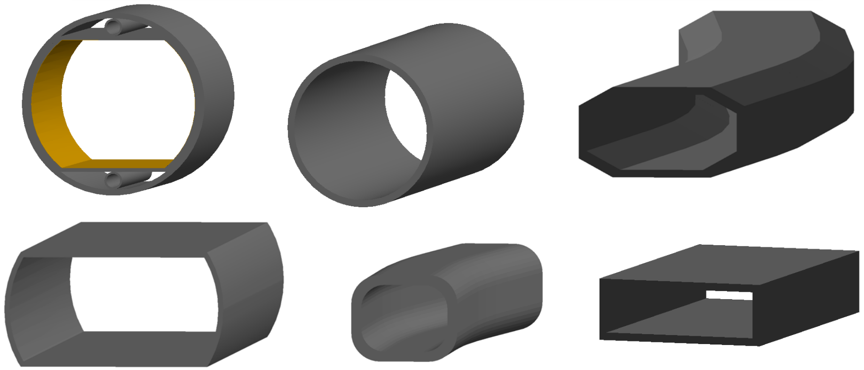
- rarely do people therefore make such a model…

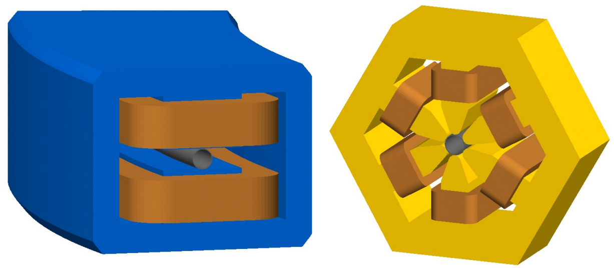
Beam Delivery Simulation (BDSIM)

BDSIM - Beam Delivery Simulation

- Create 3D Geant4 model from optical description in minutes
- Library of generic accelerator geometry in Geant4 C++
  - you can learn a lot with generic geometry
  - scalable and safe from overlaps
- MADX style input syntax in ASCII
- Can overlay other geometry and fields for more detail
- Thick lens tracking used for in-vacuum
  - replaces Geant4's 4^{th} order Runge-Kutta
- 8 different beam pipe styles
- 8 different magnet yoke styles
  - All work together dynamically







Purpose

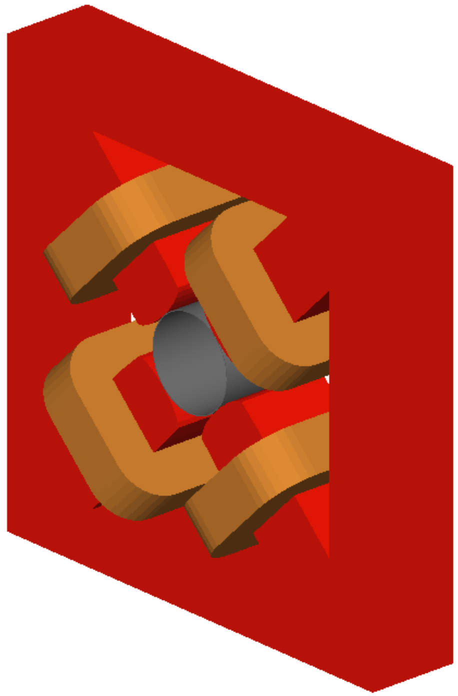
- Simulate beam loss and beam interaction with matter in a particle accelerator
- Examples:
  - transport in air (affects beam size and transmission)
  - beam degrader
  - secondary beam transport including production in the target
  - energy deposition from collimation
  - detector background
- Not intended as optical design tool
  - not a replacement for MADX of Sixtrack
  - only particle tracking -> no matrix propagation
- Prepare model from optical description

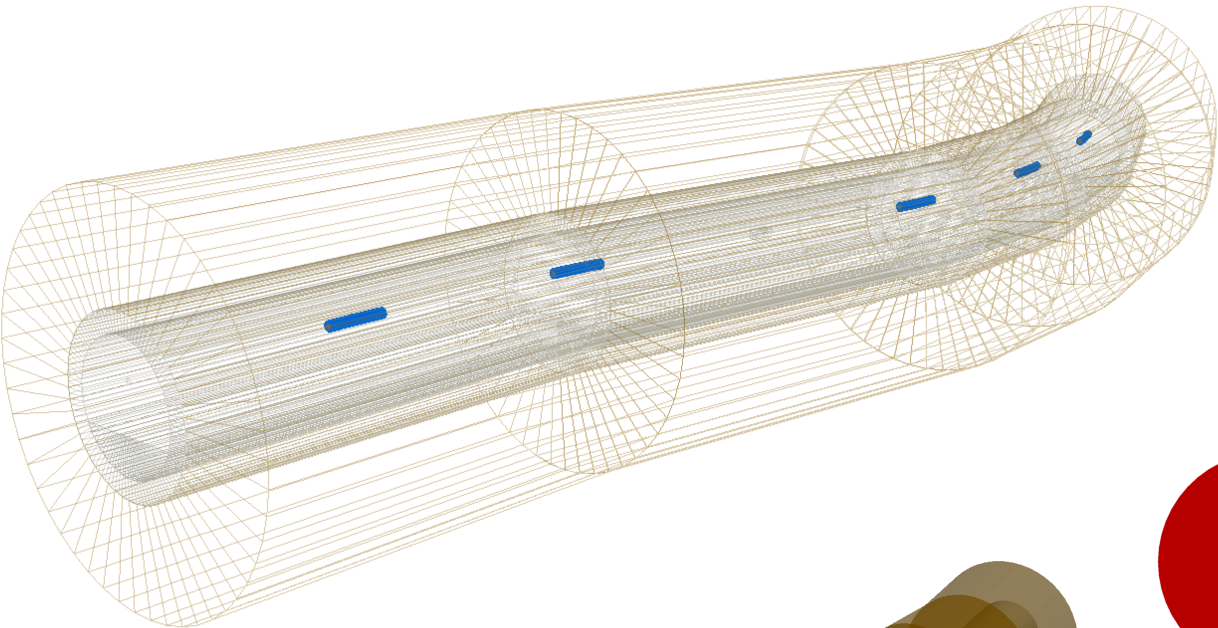
Geant4 Model Ingredients

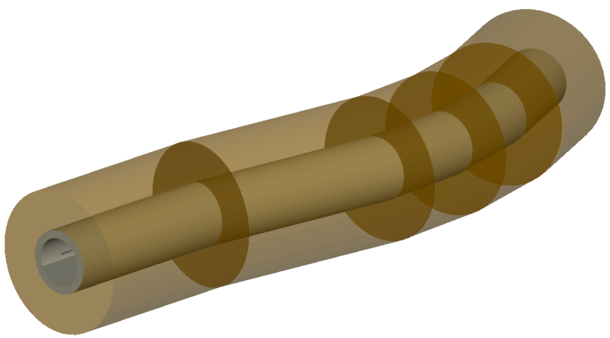
- Requires definition of
  - geometry
  - fields
  - physics processes
- Library of scalable generic geometry provided
- Matching perfect fields for each magnet provided
  - ideal multipole for yoke
- Simple interface to Geant4's modular physics lists and reference physics lists
  - modular -> "em", "ftfp_bert"
  - reference physics lists are provided by Geant4 and include several modular lists
- For accelerator tracking we provide integrators for each magnet type
  - if particle non-paraxial, we 'fall back' to a Geant4 numerical integrator (RK4)

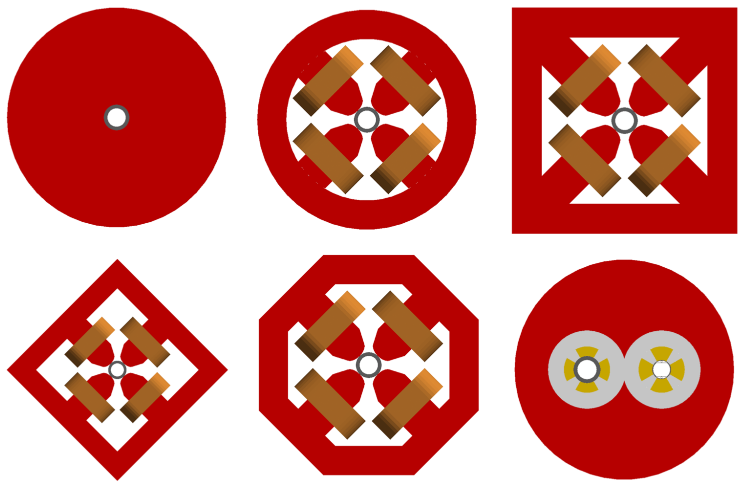
Generic Geometry

- Variety of styles for each component
  - coils included correctly even if magnet split
- Selection of generic tunnels included



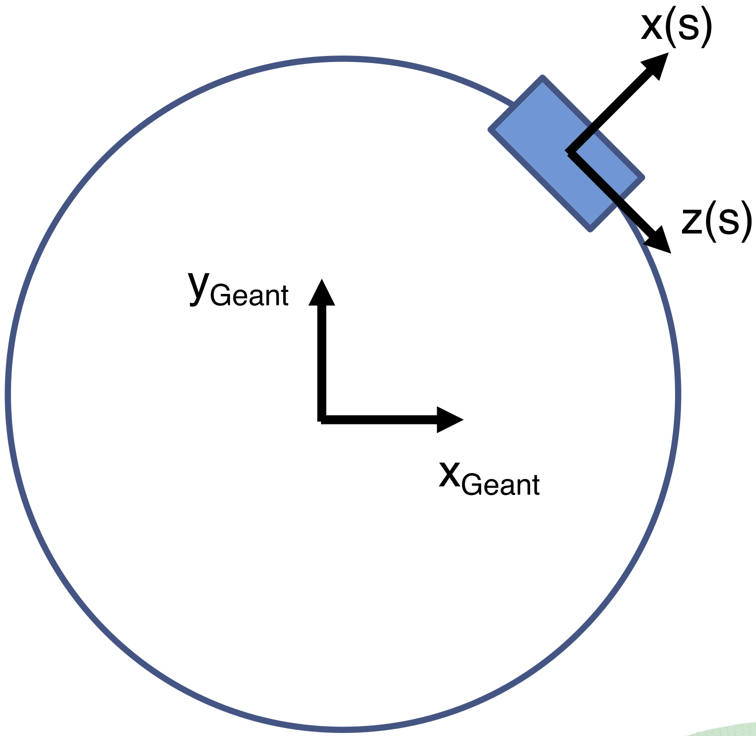


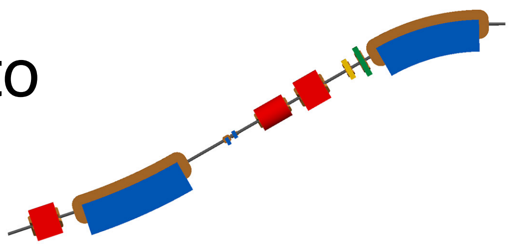


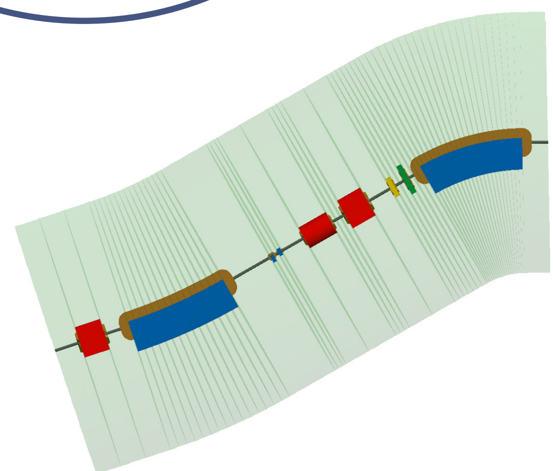


Coordinate Transforms

- Accelerator tracking uses a curvilinear system
- Geant4 uses 3D Cartesian coordinates
- Can look up transform from one volume to another
  - ie current to world (outermost)
  - level of hierarchy unknown and can vary
  - geometry may not be aligned to coordinate system
- Use *parallel* geometry to overcome this
  - different representation
- Matrix style integrators use transforms



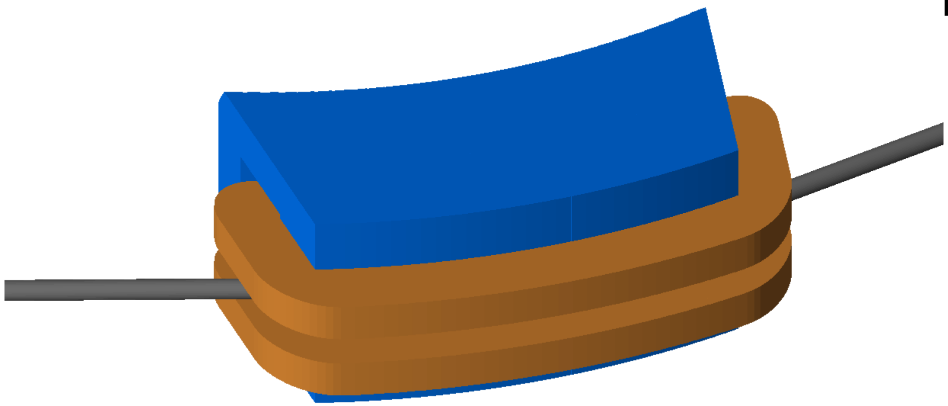




Pole Faces & Thin Elements

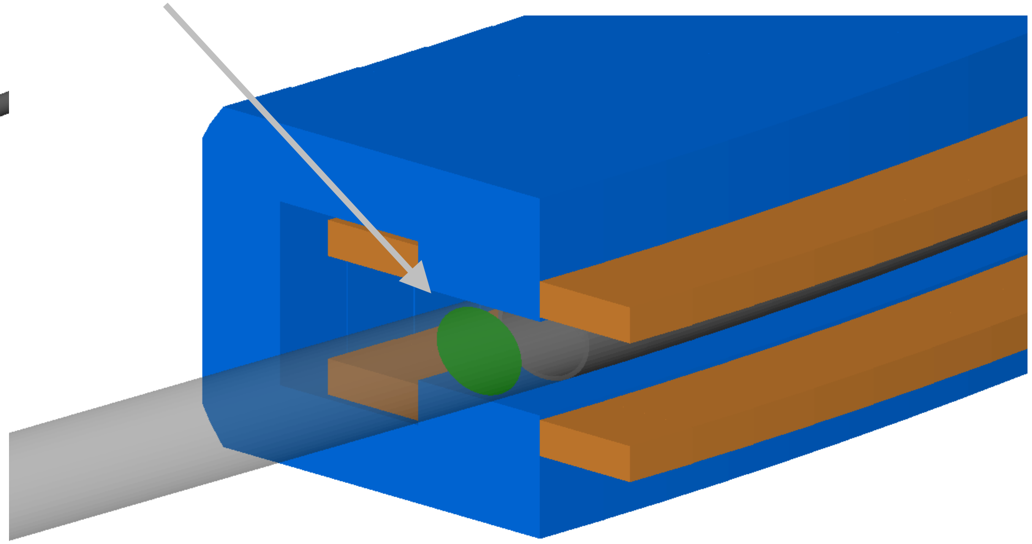
- Imperfections usually implemented via thin elements in tracking
  - entrance / exit or in the middle of magnet
- Pole face rotations contribute significantly to optics
  - crucial for low energy applications
  - Implementation using 1st order matrix formalism

Revert to Geant4 based integrator in non–paraxial limit.



Angled beam pipe and yoke geometry as well as coils

Thin element for fringe field



Example Syntax

- "GMAD" - Geant4 + MAD

sm.gmad

```
d1: drift, l=1*m;
q1: quadrupole, l=1*m, k1=1.3, magnetGeometryType="polesfacet";
c1: rcol, l=0.6*m, ysize=5*mm, xsize=5*mm, material="Copper", outerDiameter=10*cm;
s1: sbend, l=1*m, angle=0.10;

l1: line = (d1, q1, d1, c1, d1, s1);
use,period=l1;

sample, all;

option, ngenerate=1,
        physicsList="em";

beam, particle="proton",
      energy=10.0*GeV,
      X0=0.001,
      Y0=0.001;
```

```
-:---  sm.gmad        All L3    Git:develop  (Fundamental)
Wrote /Users/nevay/physics/reps/bdsim/examples/simpleMachine/sm.gmad
```

Model Conversion

- BDSIM uses MAD(8,X) style syntax
- Can write manually, but can also convert easily
- Prepare 'flat' optical description of lattice
  - here prepare MADX TFS format Twiss table

```
select,flag=twiss, clear;
twiss,sequence=SEQUENCENAME, file=twiss.tfs;
```

- Convert using pybdsim Python utility

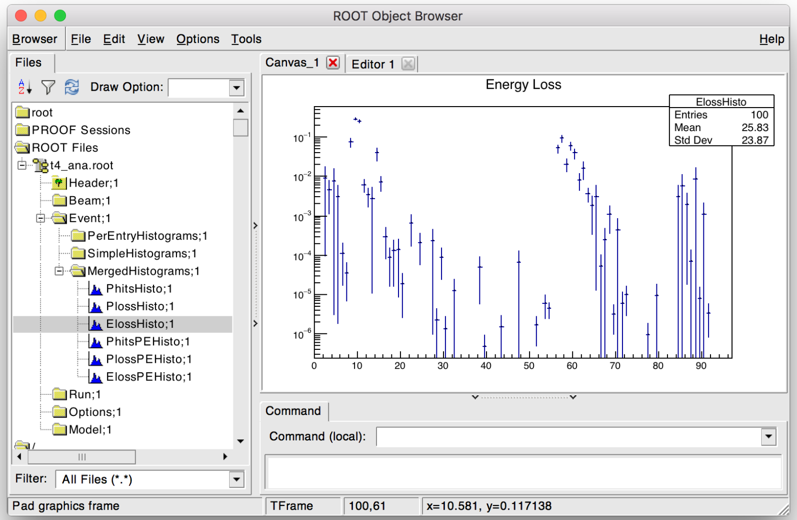
```
>>> a,b,c = pybdsim.Convert.MadxTfs2Gmad('inputfile.tfs', 'latticev1')
```

- Fold in information by name - Python dictionaries
  - up to user to how this information is sourced

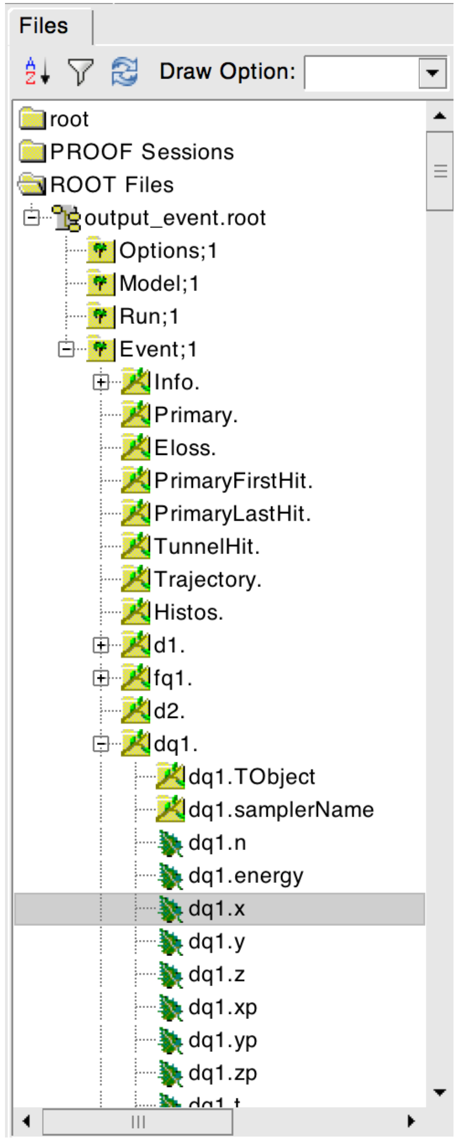
```
>>> drift123dict = {'aper1':0.03, 'aper2':0.05, 'apertureType':'rectangular'}
>>> quaddict = {'magnetGeometryType':'polesfacetcrop}
>>> d = {'drift123':drift123dict, 'qf1x':quaddict}
>>> a,o = pybdsim.Convert.MadxTfs2Gmad('inputfile.tfs', 'latticev1', userdict=d)
```

Output

- Use ROOT format for data
  - highly suited to particle physics event by event storage and analysis
- Well documented and widely used
  - support + community
- Scales well to large / parallel data sets
- Specifically designed for data evolution
- BDSIM is strongly reproducible
  - all seeds and settings stored



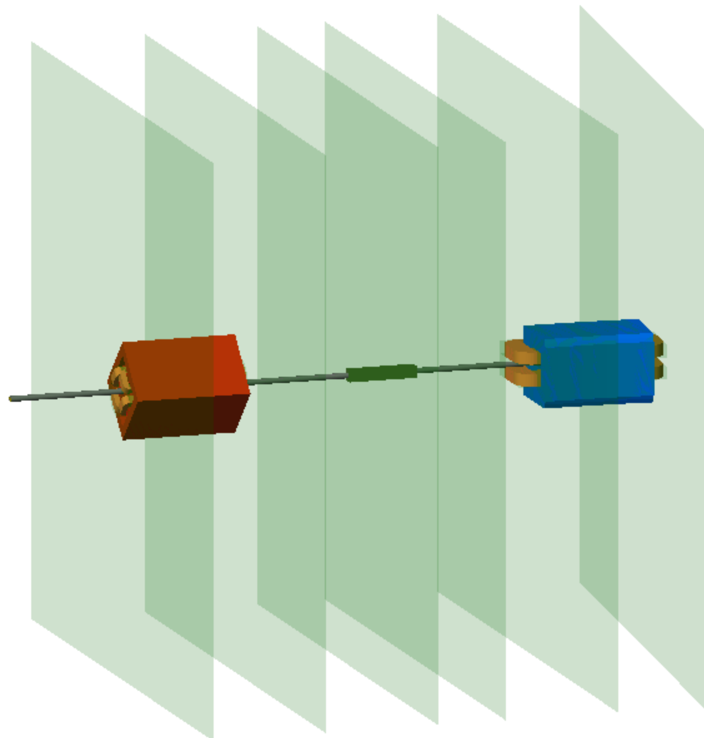
structure of output file



Information Reduction

- particle physics simulation produces huge potential amount of information
  - coordinates of every step of every particle…
- Geant4 is 'silent' by default
  - developer chooses ➞ record what's key
- Energy deposition recorded by default
- Optional samplers
  - plane after each element that records all particle
- Optional trajectories
  - record 'history' or particles of interest
- Event by event storage
  - unlike tracking code, not 1 particle : 1 event
  - crucial for correct statistical uncertainties

sampling planes after each element (normally invisible)



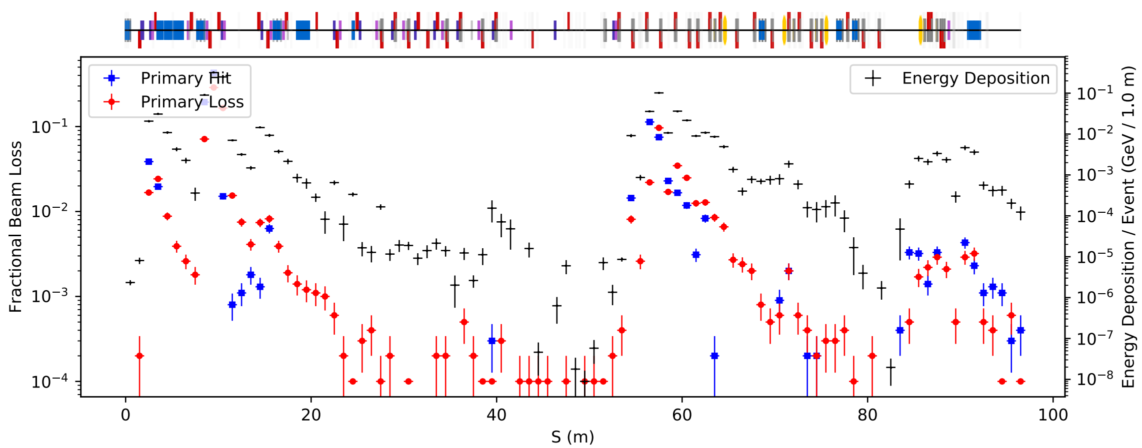
Analysis

- Analysis tool 'rebdsim' (root event BDSIM)
- Event by event analysis
- e.g. all neutrons over 20GeV that interact with collimator
  - no problem!
- Simple text input for 1,2,3D histograms

analysisConfig.txt

```
InputFilePath       ../../data/sample1.root
OutputFileName      ./ana1.root
# Object        treeName    Histogram Name      # Bins      Binning                     Variable                Selection
Histogram1D     Event.      EnergyLoss          {22}        {0:22}                      Eloss.S                 Eloss.energy*Eloss.weight
Histogram2D     Event.      Flux                {30,30}     {-2.5:2.5,-2.5:2.5}         fq1.y:fq1.x             fq1.energy*fq1.weight
Histogram3D     Event.      Flux3D              {30,30,100} {-2.5:2.5,-2.5:2.5,0:1}     fq1.x:fq1.y:fq1.energy  1
Histogram2D     Event.      Neutrons            {50,50}     {-2:2,-2:2}                 fq1.y:fq2.x             fq1.weight*(fq1.partID==2212)
```

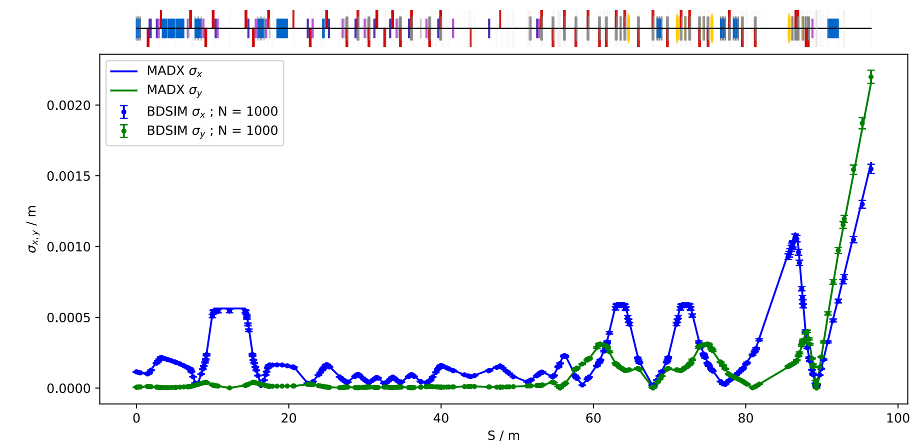
'selection' is weighting
can be Boolean and or numerical

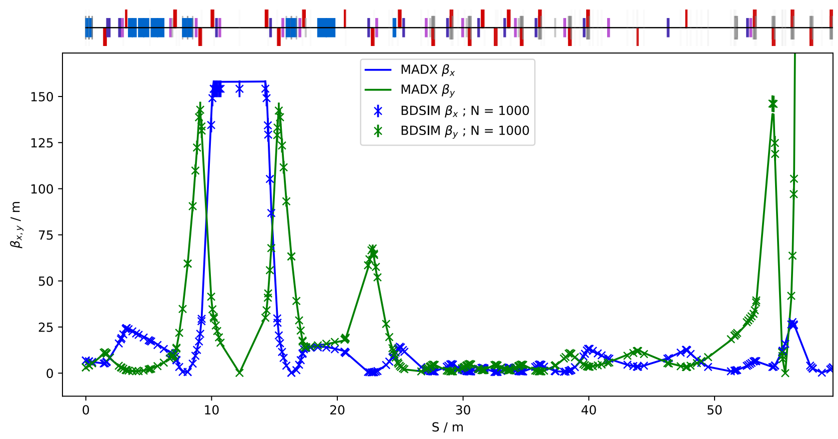


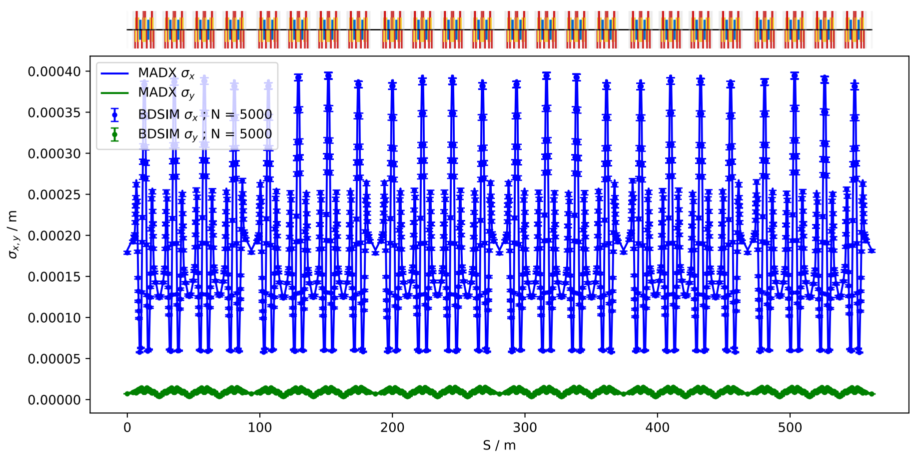
basic impact, loss and
energy deposition analysis

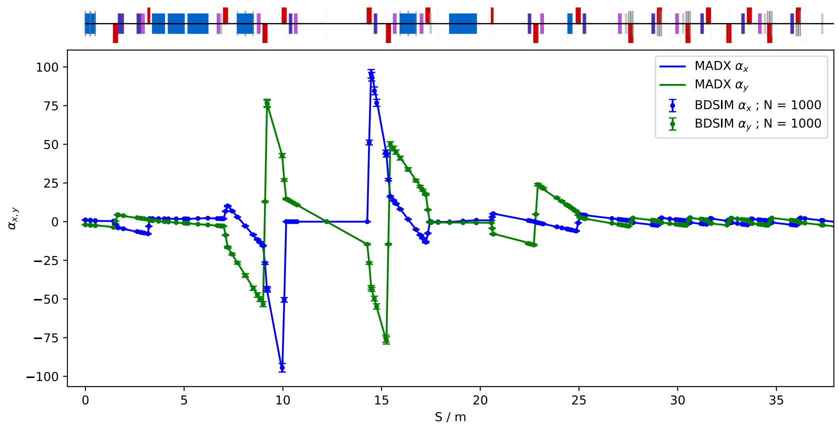
Optical Function Comparison

- Particle distribution recorded after each element
- Calculate optical functions from particle distribution
  - using (up to) 4th order moments
  - full statistical uncertainty calculated too





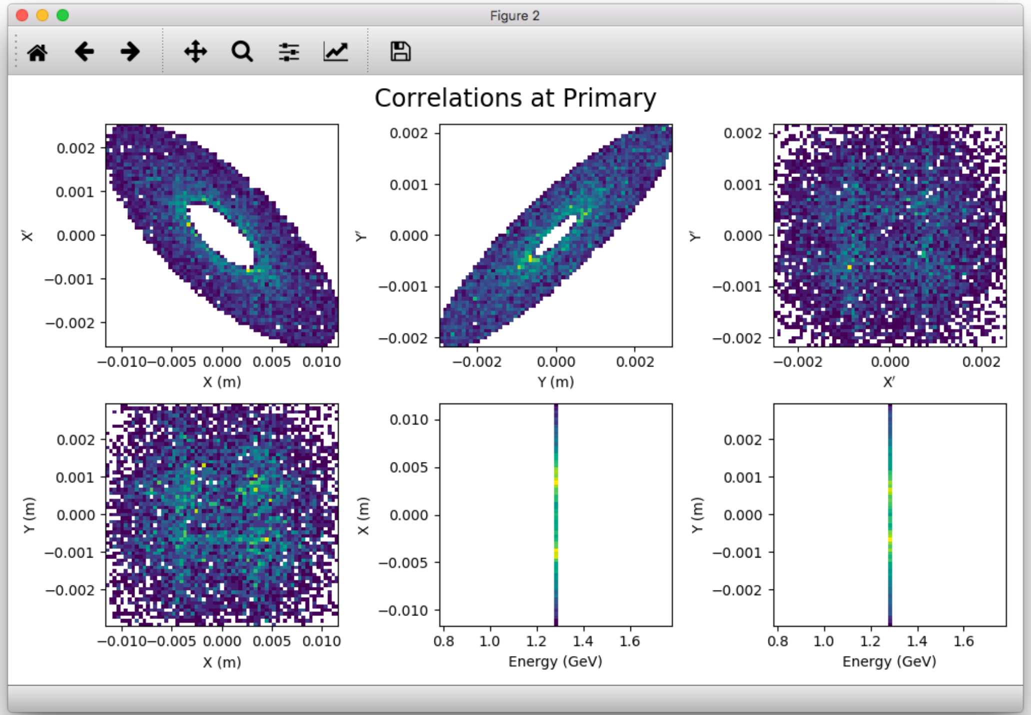


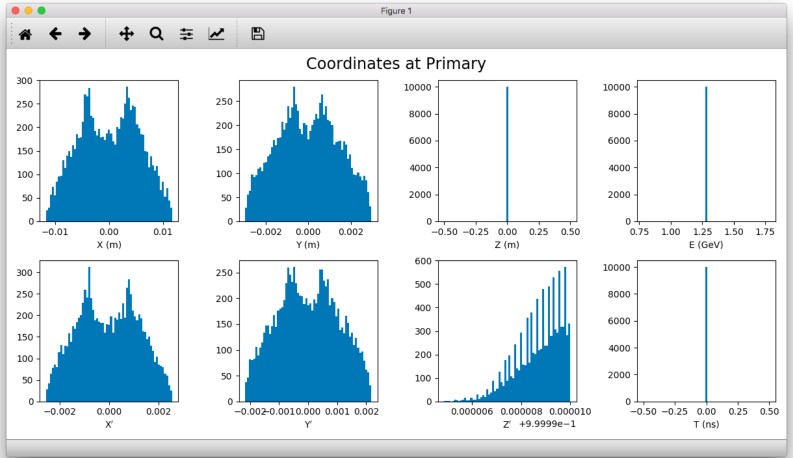


Adding more detail…

Beam Distribution

- Beam interaction rare - need more efficient simulation
- Interaction at 50 sigma?
- Efficiently generate required distribution

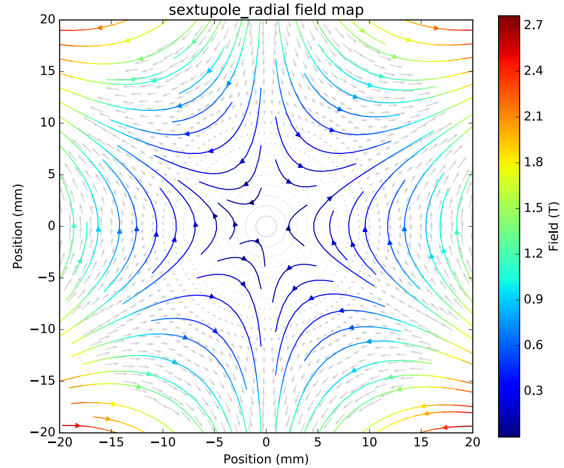




variety of distributions included using CLHEP pseudo-random number generator

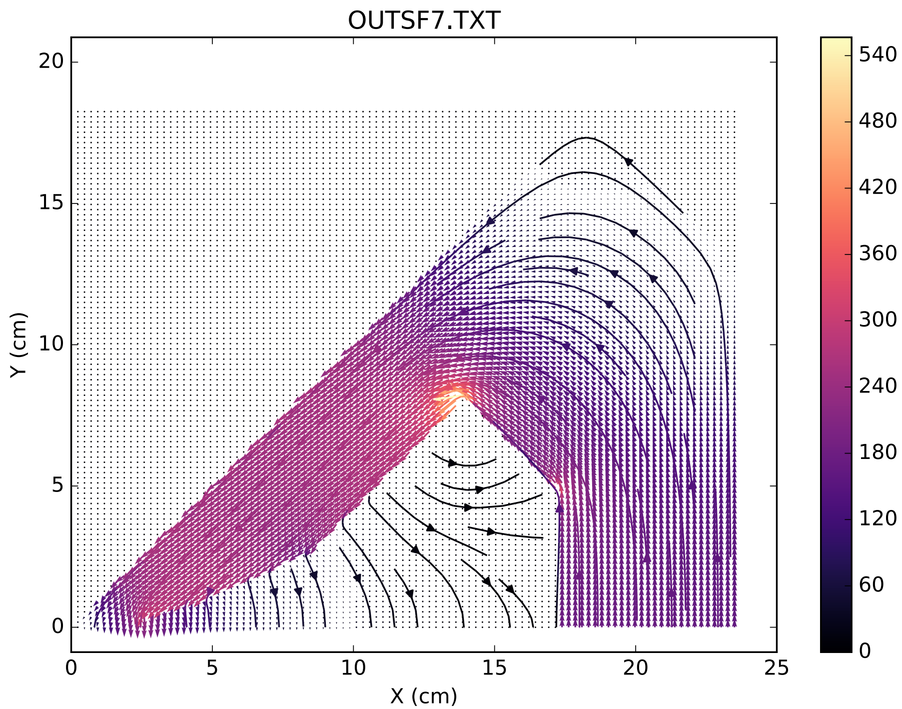
Field Maps

- Equations describe pure fields
- Can overlay field map on BDSIM generic element
  - yoke or vacuum separately or both together
- 1- 4D loading and interpolation
  - nearest neighbour, linear and cubic interpolation



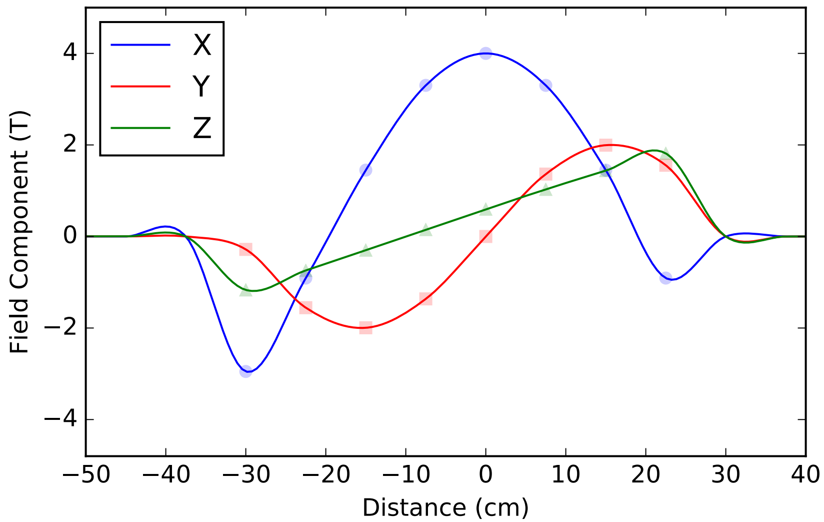
ideal sextupole field

$$B_x = \frac{1}{2!}\frac{\partial^2 B_y}{\partial x^2}\,2xy$$
$$B_y = \frac{1}{2!}\frac{\partial^2 B_y}{\partial x^2}\,(x^2 - y^2)$$



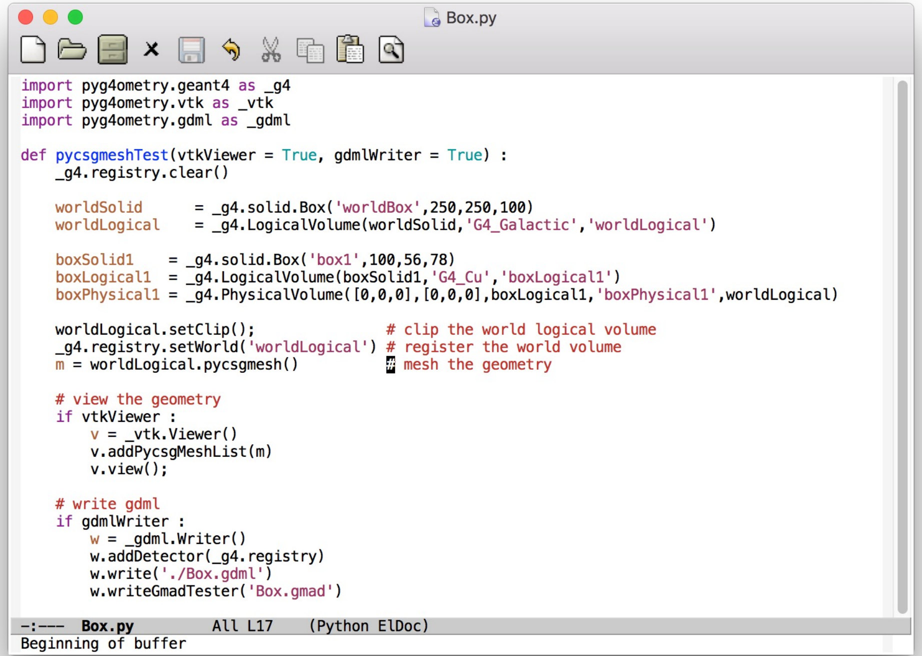
example field map from Poisson 2D

example interpolation



Geometry Preparation

- More detailed geometry required for some specific components
- Can easily load GDML format geometry
  - XML format of Geant4
- pyg4ometry package to create GDML in a script
  - the same as Geant4 C++ geometry but in Python
- Visualise using VTK library
  - identify overlaps
- Shrink container volume to fit contents
- Being actively developed

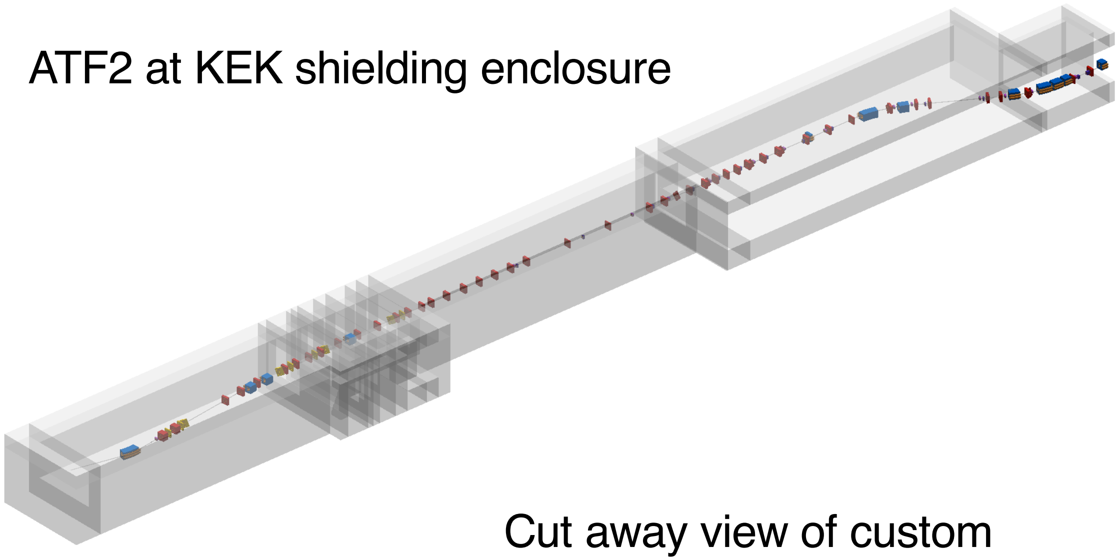



```python
import pyg4ometry.geant4 as _g4
import pyg4ometry.vtk as _vtk
import pyg4ometry.gdml as _gdml

def pycsgmeshTest(vtkViewer = True, gdmlWriter = True) :
    _g4.registry.clear()

    worldSolid      = _g4.solid.Box('worldBox',250,250,100)
    worldLogical    = _g4.LogicalVolume(worldSolid,'G4_Galactic','worldLogical')

    boxSolid1    = _g4.solid.Box('box1',100,56,78)
    boxLogical1  = _g4.LogicalVolume(boxSolid1,'G4_Cu','boxLogical1')
    boxPhysical1 = _g4.PhysicalVolume([0,0,0],[0,0,0],boxLogical1,'boxPhysical1',worldLogical)

    worldLogical.setClip();                    # clip the world logical volume
    _g4.registry.setWorld('worldLogical') # register the world volume
    m = worldLogical.pycsgmesh()           # mesh the geometry

    # view the geometry
    if vtkViewer :
        v = _vtk.Viewer()
        v.addPycsgMeshList(m)
        v.view();

    # write gdml
    if gdmlWriter :
        w = _gdml.Writer()
        w.addDetector(_g4.registry)
        w.write('./Box.gdml')
        w.writeGmadTester('Box.gmad')
```

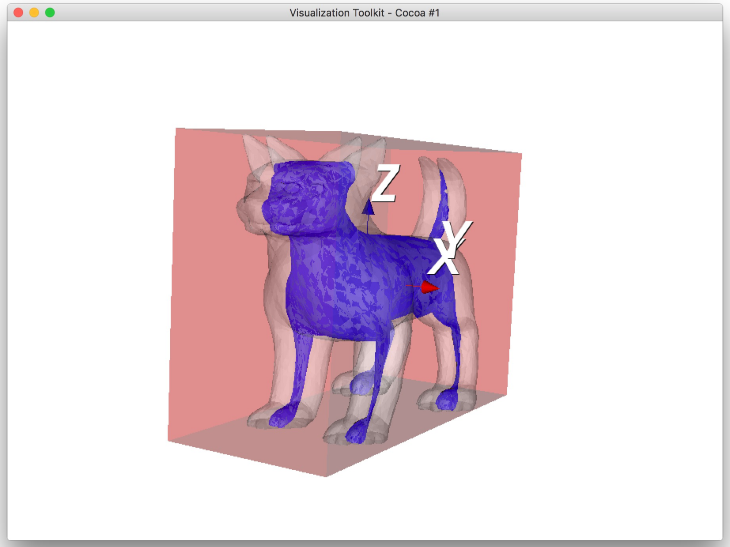
```
-:---  Box.py        All L17     (Python ElDoc)
Beginning of buffer
```

Geometry Examples

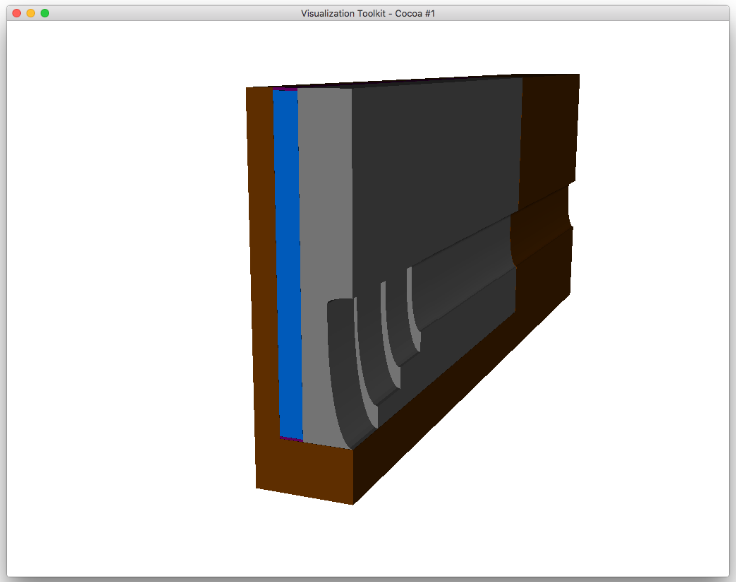
ATF2 at KEK shielding enclosure

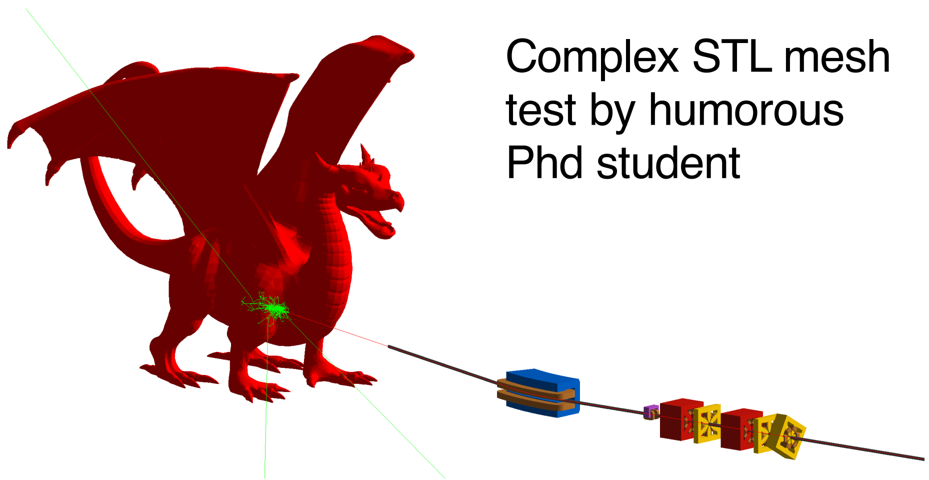


Overlap identification with complex STL mesh



Cut away view of custom collimator for CLIC



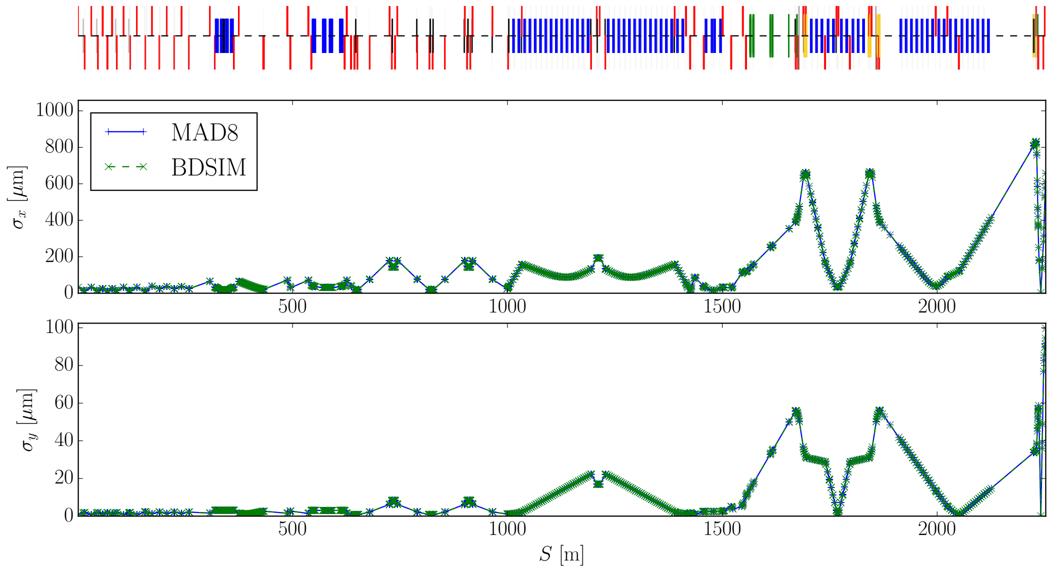


Complex STL mesh test by humorous Phd student

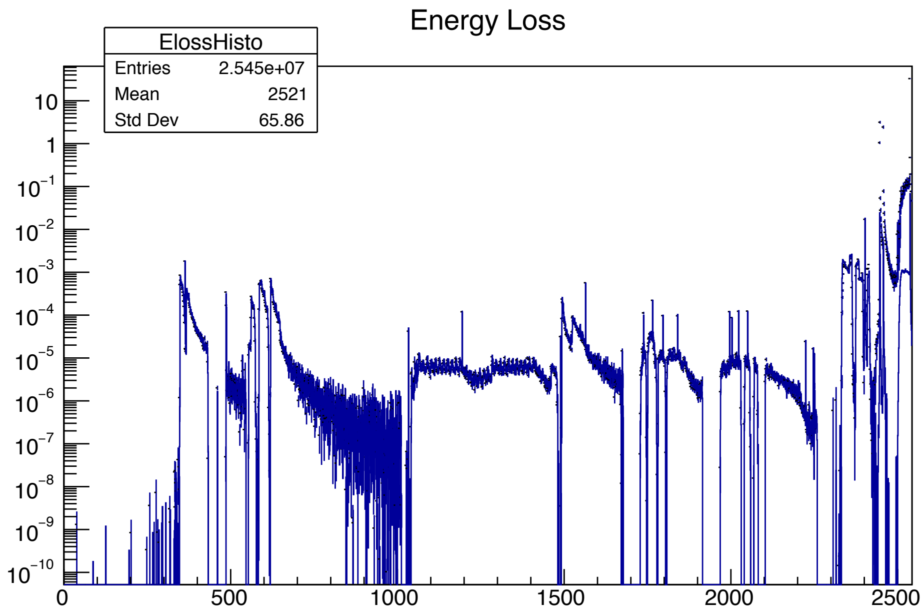
Examples

ILC 250 GeV Model

- Beam delivery system from BSY to IP
  - Linear optics agree well
  - Collimation defined by *no synchrotron radiation hitting final doublets*
  - What about all the losses long the BDS

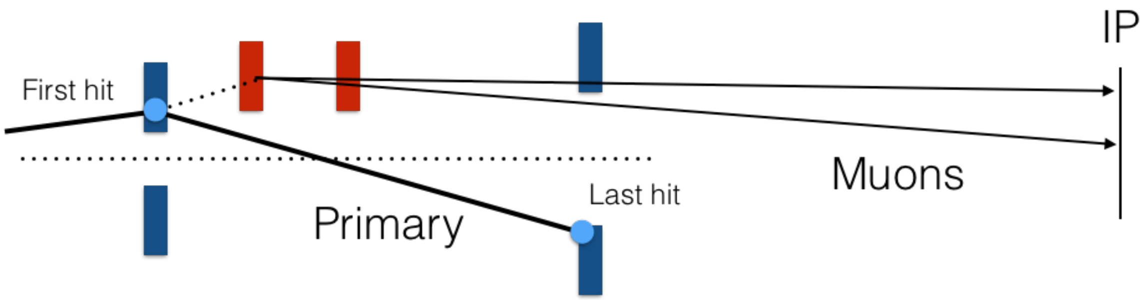


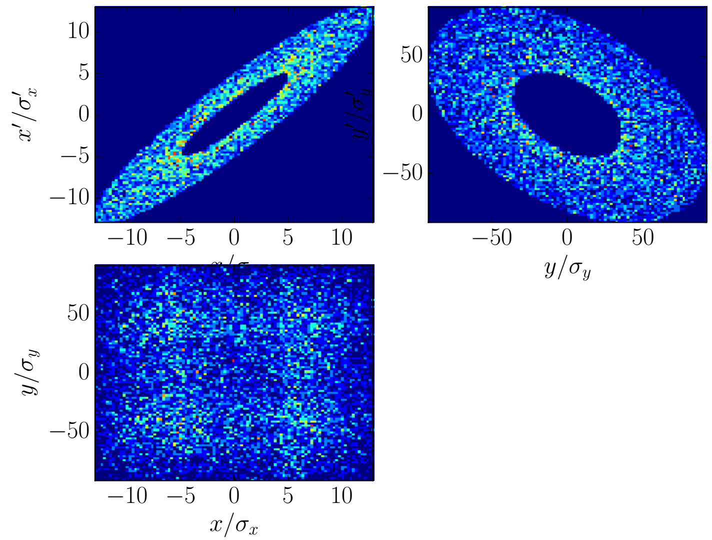
- Synchrotron radiation
  - Simple test of photon emission from all magnetic elements
  - Uses well tested and built in Geant4 SR model

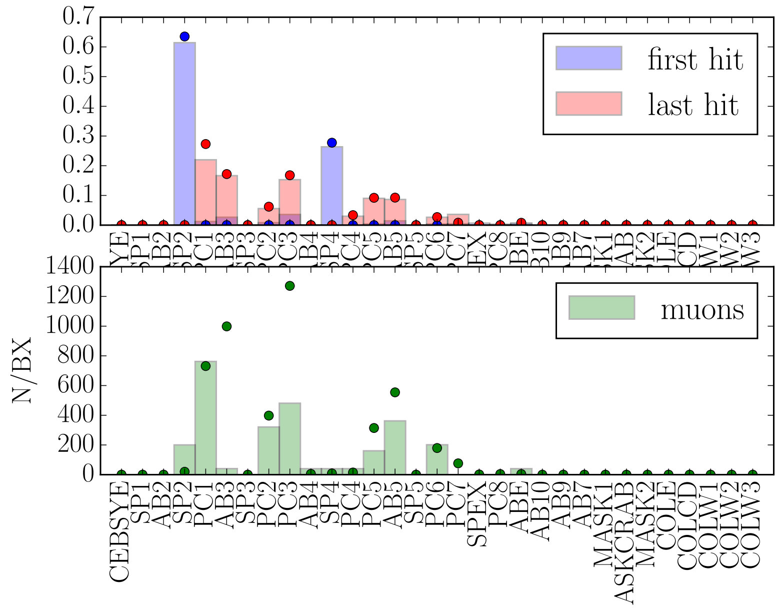


S. Boogert

ILC Muon Production Example





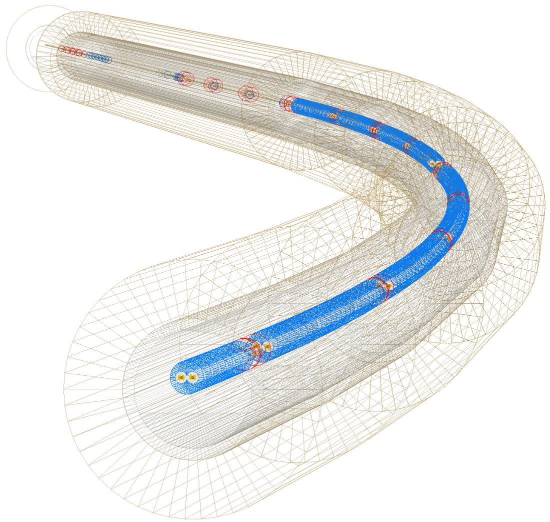


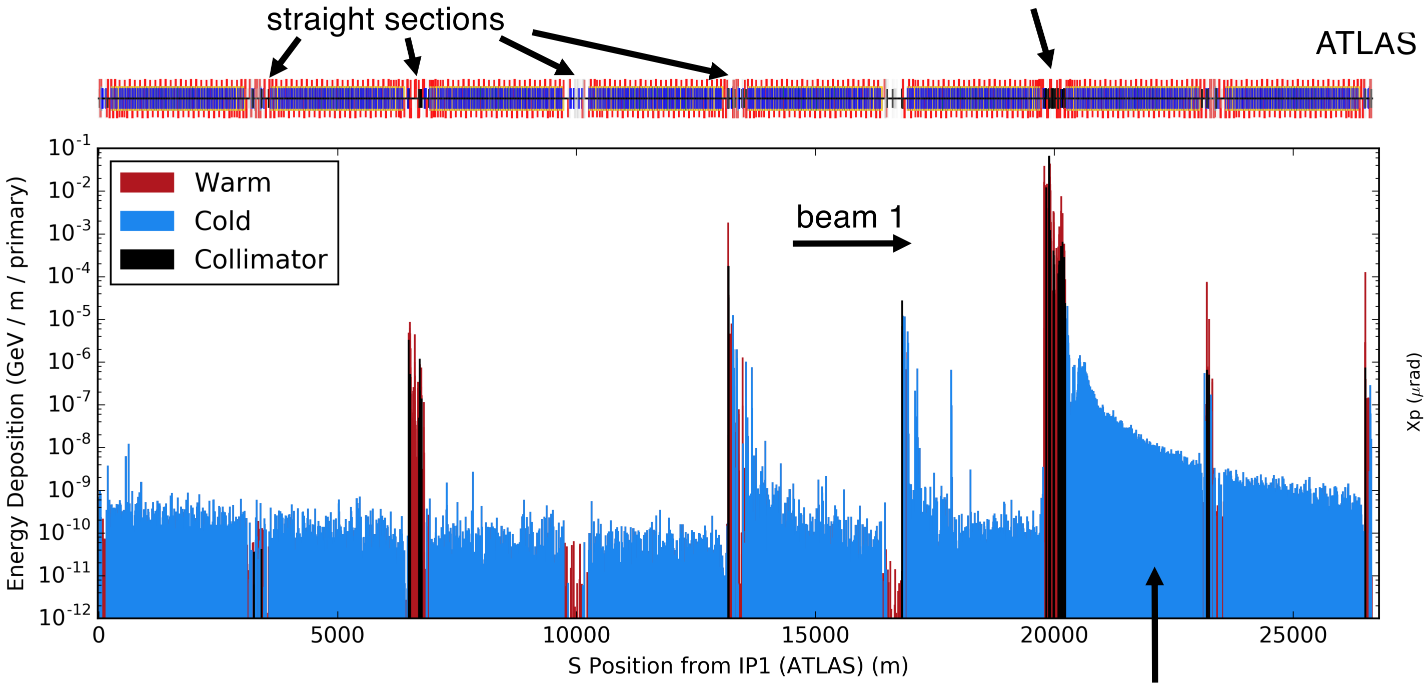
- ILC muon production interesting as large distance between production point and IP
  - Interaction in collimation system produces large number of muons
- Halo
  - 2×10^{10} electrons per bunch
  - Halo is 1×10^{-3} of total beam
  - $1/\varepsilon_{SP}$ distribution
  - x : 5 – 13 σ
  - y : 36 – 92 σ

S. Boogert

LHC Collimation

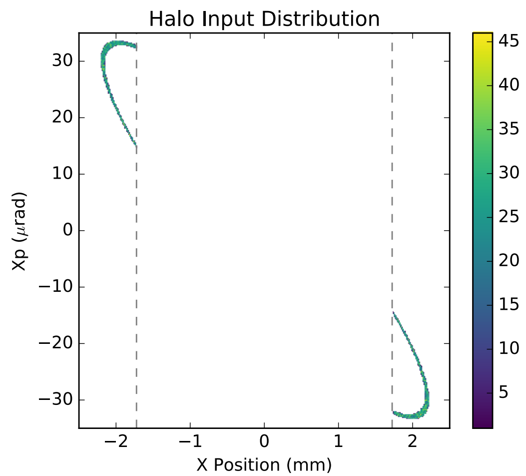
- Halo populated during beam storage
- Continually removed
- Simulate halo as it touches collimators
- LHC-style dipoles & quadrupoles
- Require 1:10^6 precision





B1 4TeV energy deposition map

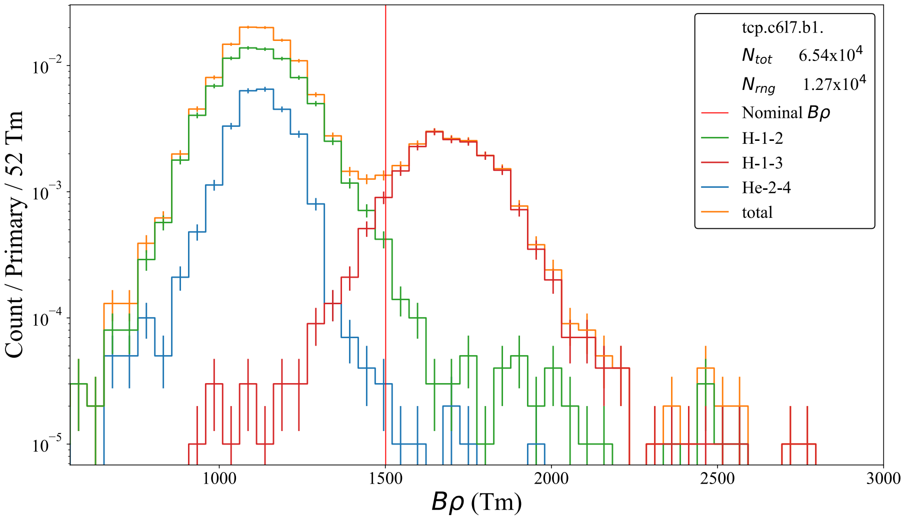
Example halo distribution

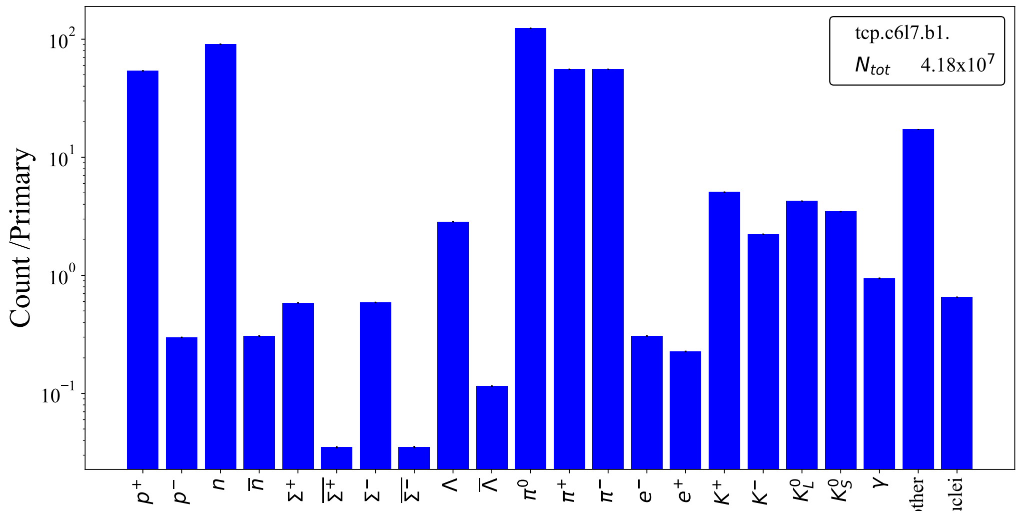


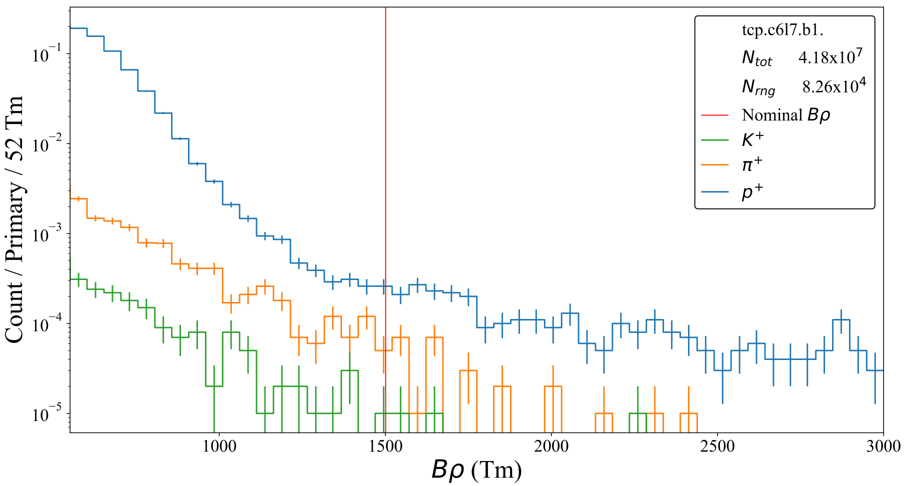
L. Nevay, S. Boogert

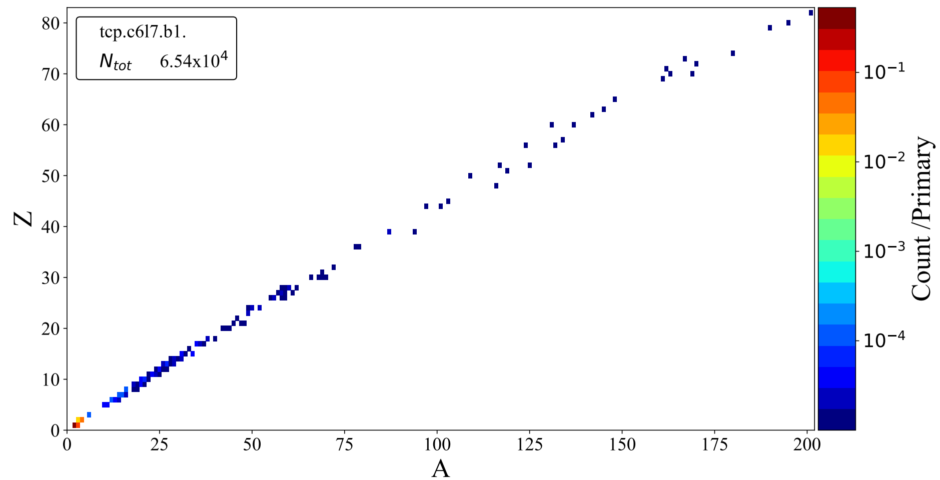
LHC Ion Collimation

- Similarly, same model can be used with ions
- Fragmentation - many fragments around nominal B***ρ***





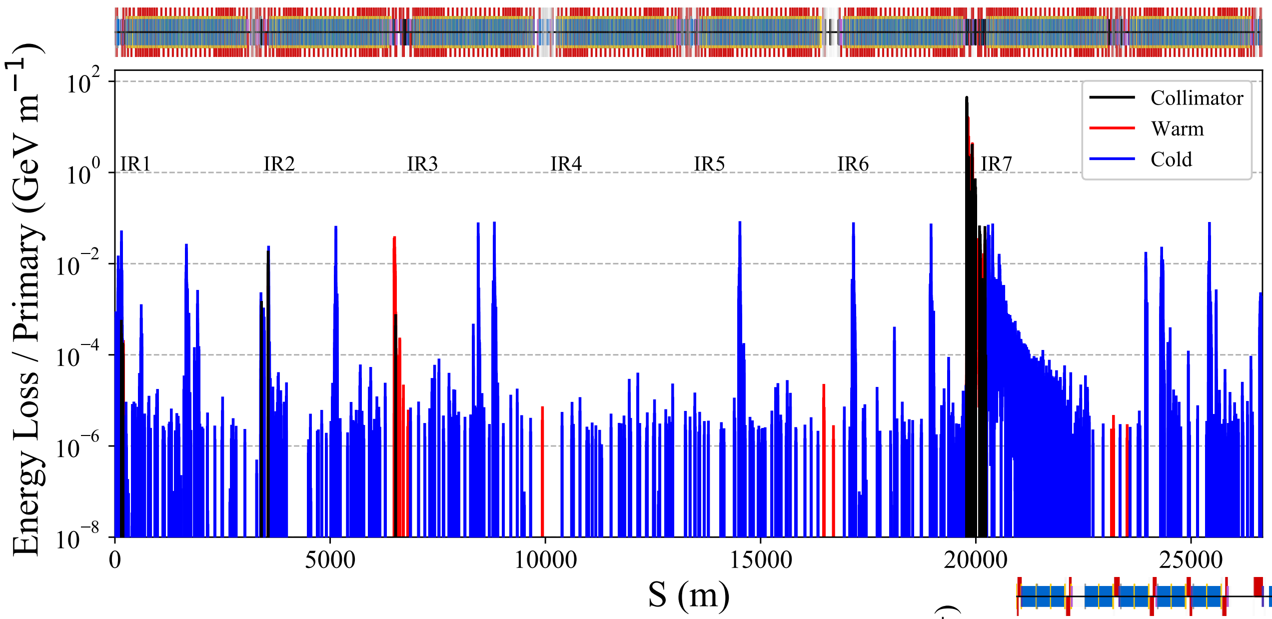




A. Abramov

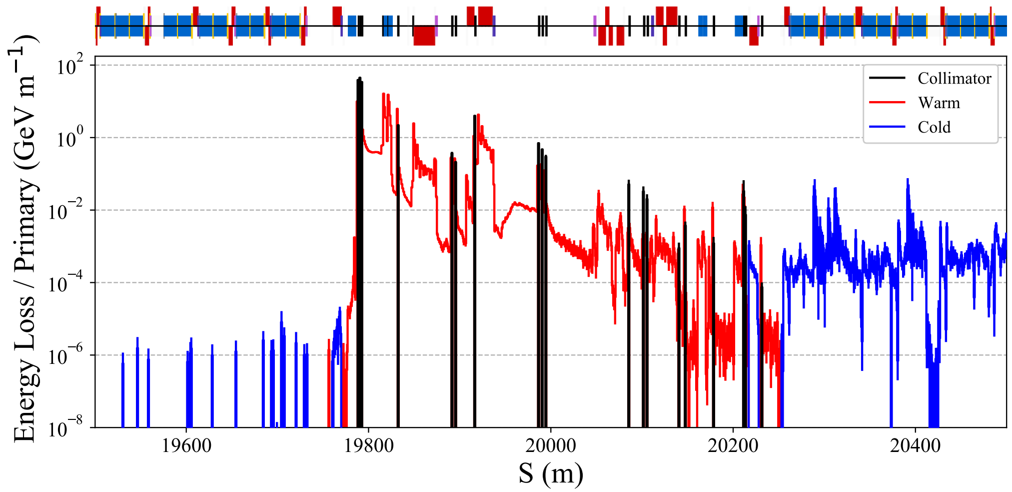
LHC Ion Collimation II

- Energy deposition around ring



- Significantly more loss spikes around ring
- Beam intensity limit much lower
- Collimator impacts only at S = 20000m here

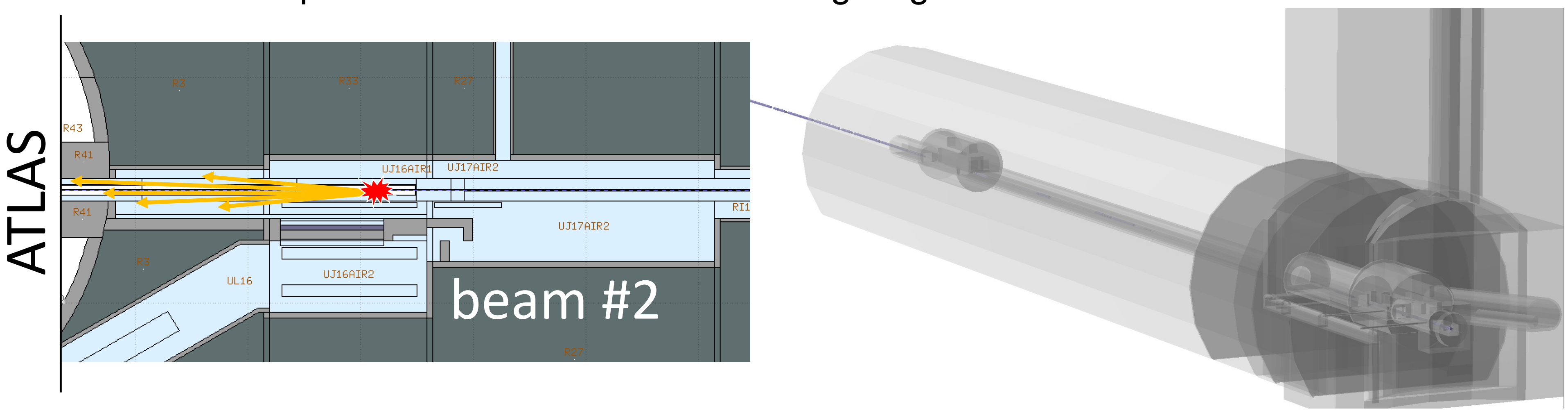
- Zoom of collimation section ('IR7')
- Coded losses on collimators, warm and cold sections



A. Abramov

LHC Non-Collision Backgrounds

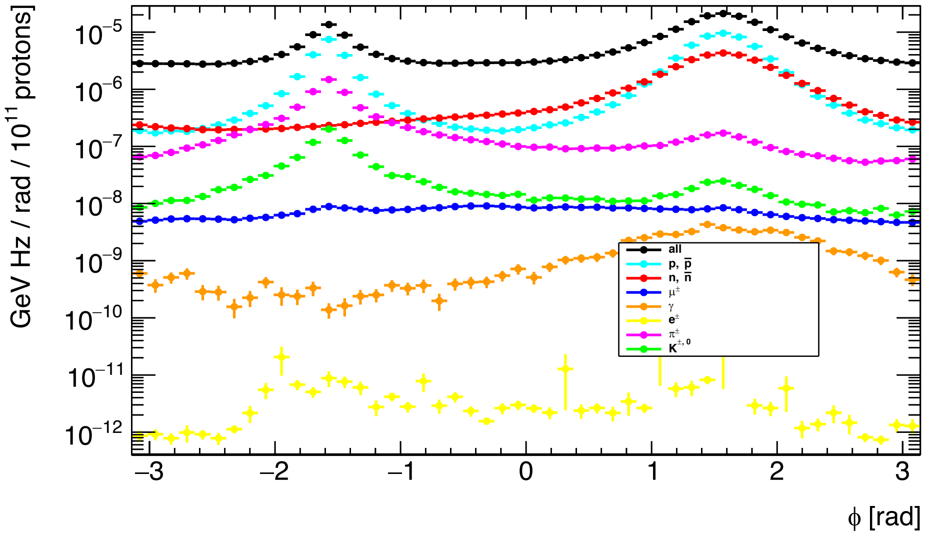
- Interaction with residual vacuum creates measurable background in ATLAS and CMS detectors
- Modelling ATLAS background using BDSIM
  - last 500m of machine before ATLAS
  - single pass simulation
  - predict observed rates in pixel detector
  - IR1 tunnel model converted from FLUKA
- Bias proton inelastic scattering with residual vacuum
  - subsequent interactions with normal weighting



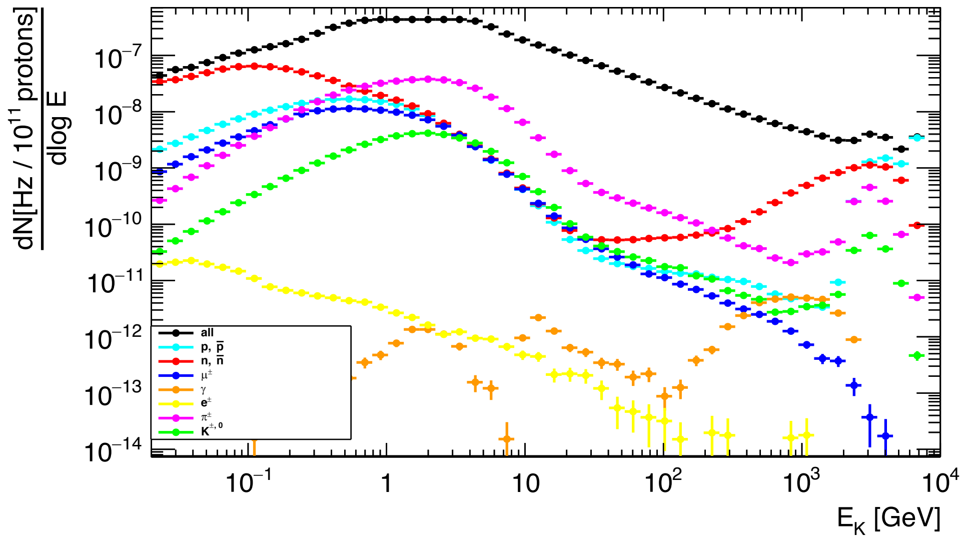
S. Walker, S. Gibson

LHC Non-Collision Backgrounds III

Azimuthal rate for different species

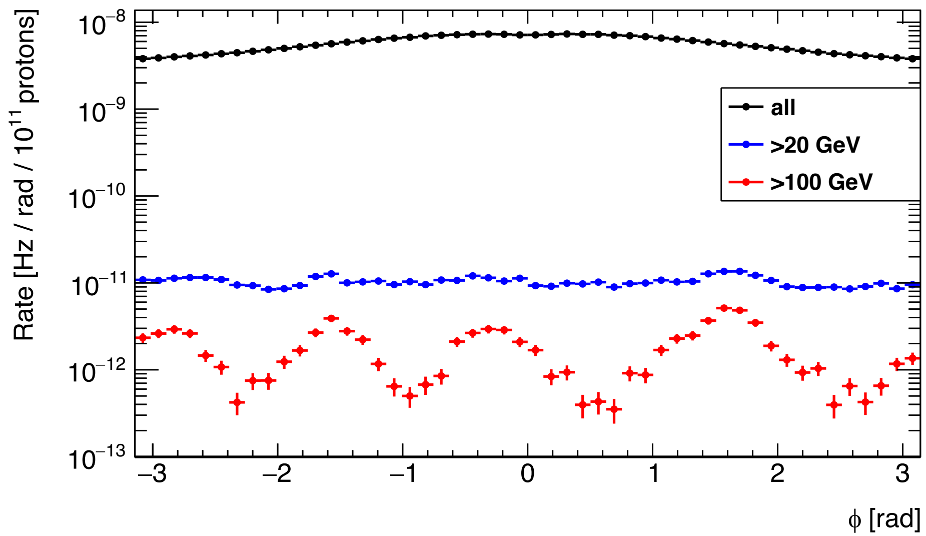


- Particles recorded at 'interface plane'
  - start of detector cavern
- Transferred to dedicated ATLAS simulation



Overall particle spectra at interface plane

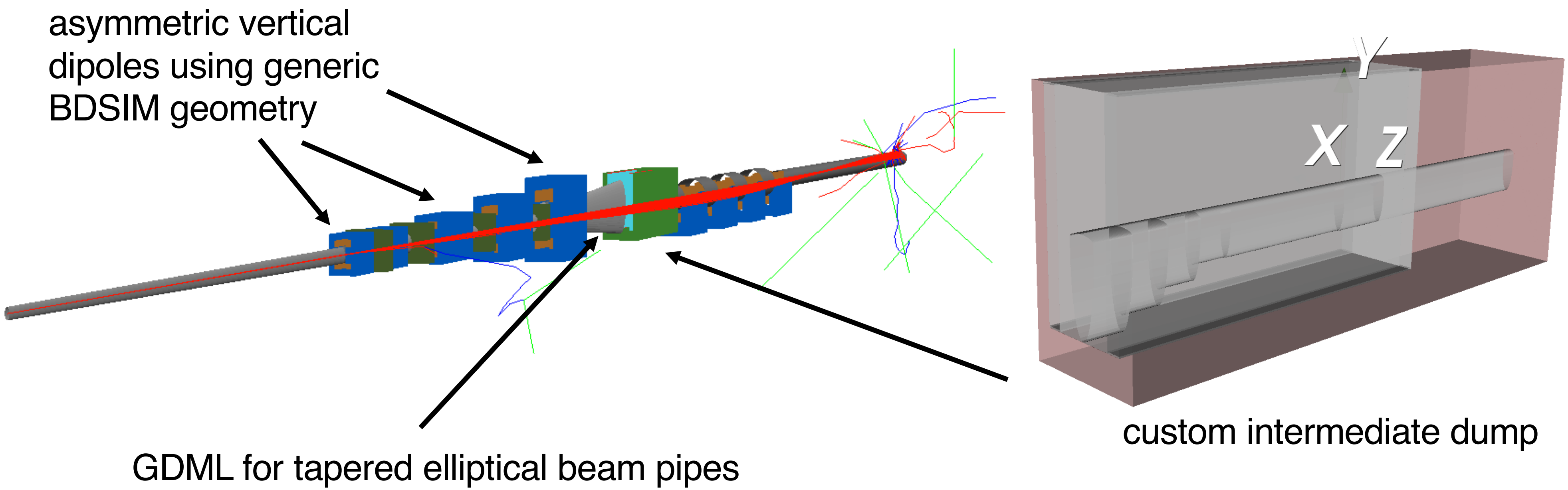
Azimuthal rate for different muon energies



S. Walker, S. Gibson

CLIC Post Collision Line

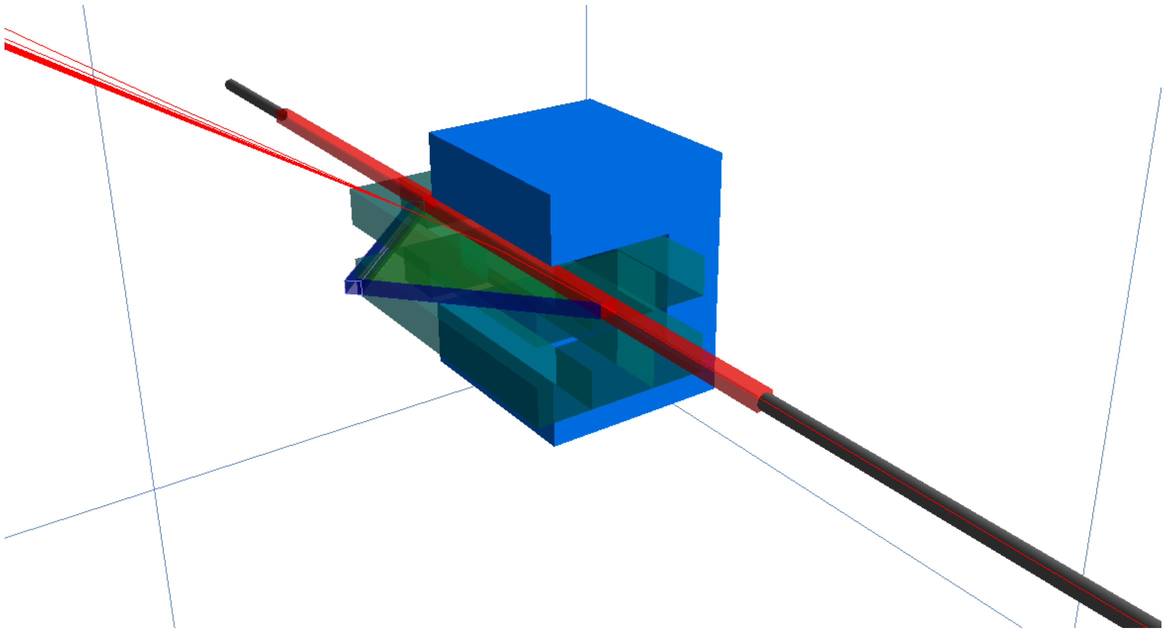
- Validate design for new proposed energy points
- Highly disrupted post collision beam
  - simulated using GUINEA-PIG
- Synchrotron radiation significant
  - leads to 2 separate beams on the dump
- Intermediate dump built using pyg4ometry package



R. Bodenstein, A. Abramov

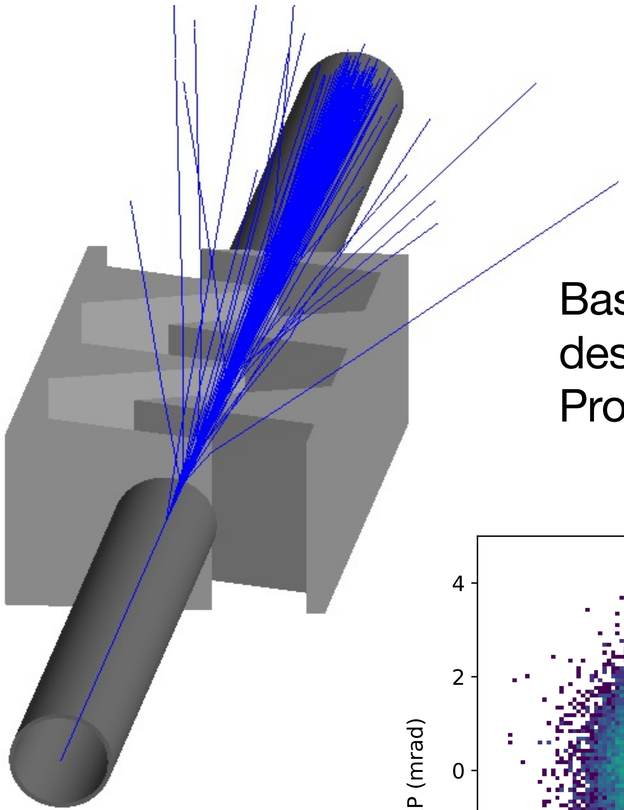
AWAKE Dipole Spectrometer

- Previous developer of BDISM L. Deacon in AWAKE collaboration
- AWAKE dipole spectrometer added to BDSIM
  - multi-layered scintillator screen
- Recently used for the calibration of the dipole
- https://www.nature.com/articles/s41586-018-0485-4

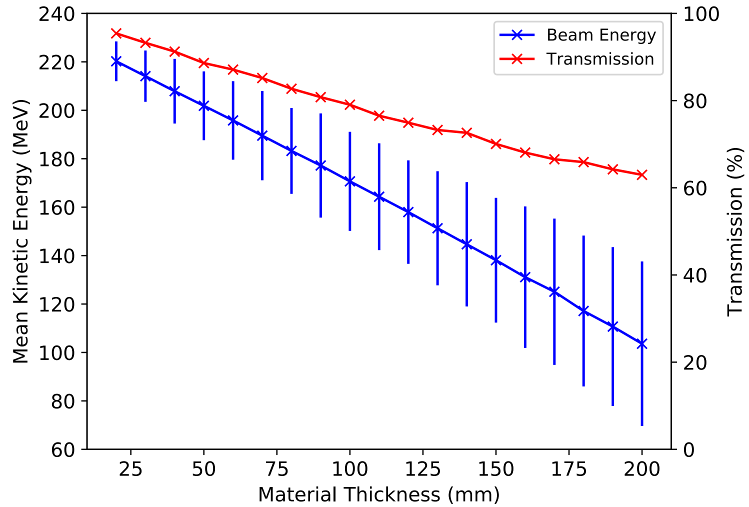


Hadron Therapy Degrader

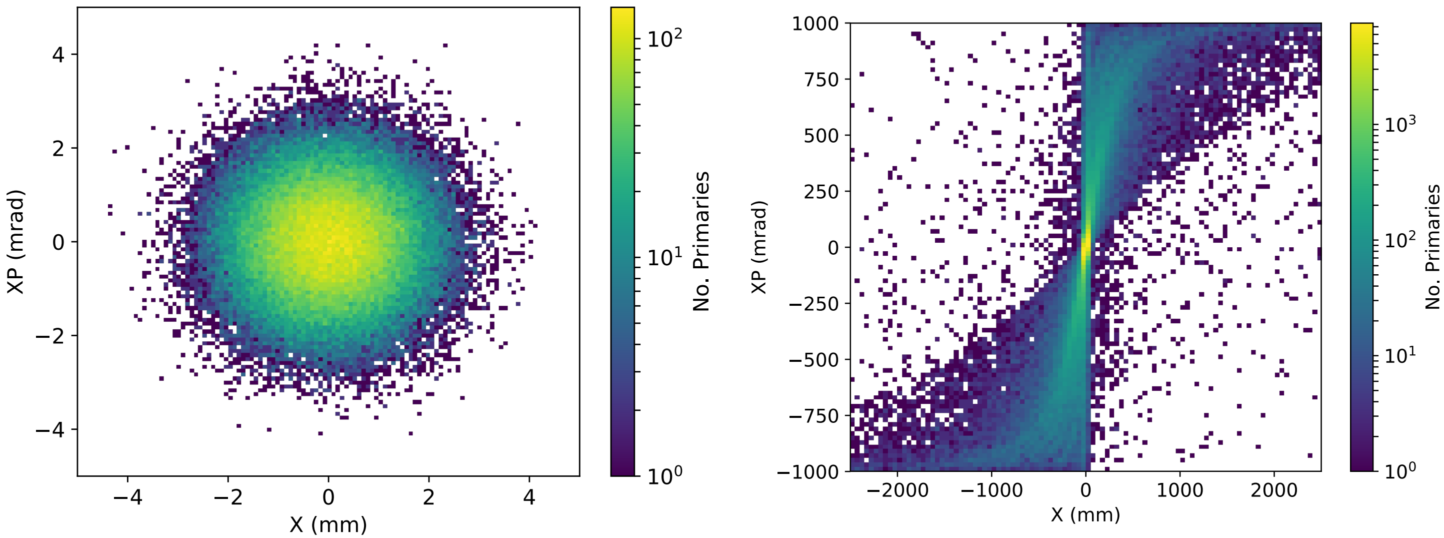
- Use variable material depth to degrade beam energy



Based on the degrader design at the Center for Proton Therapy at PSI.

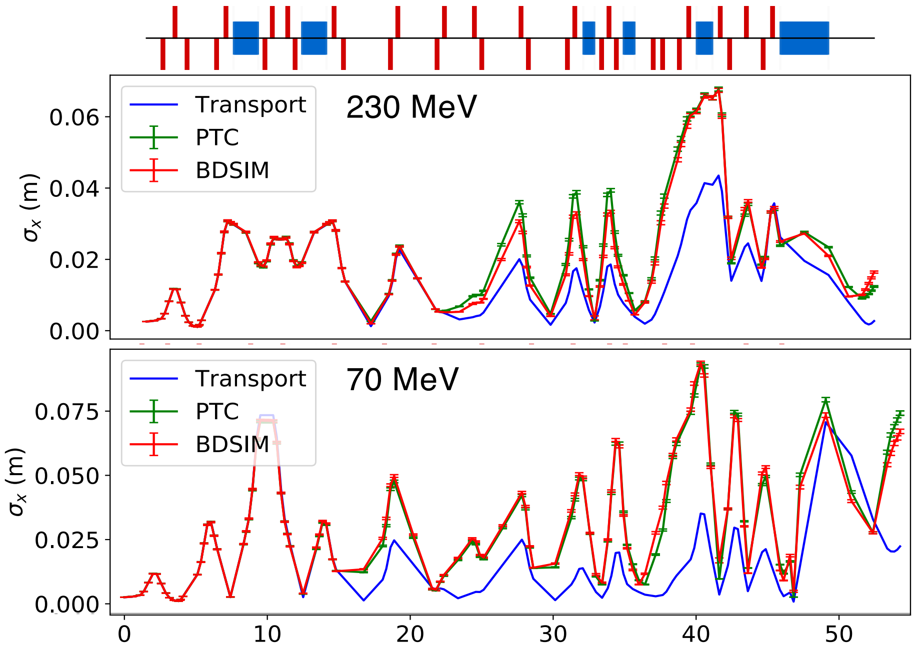


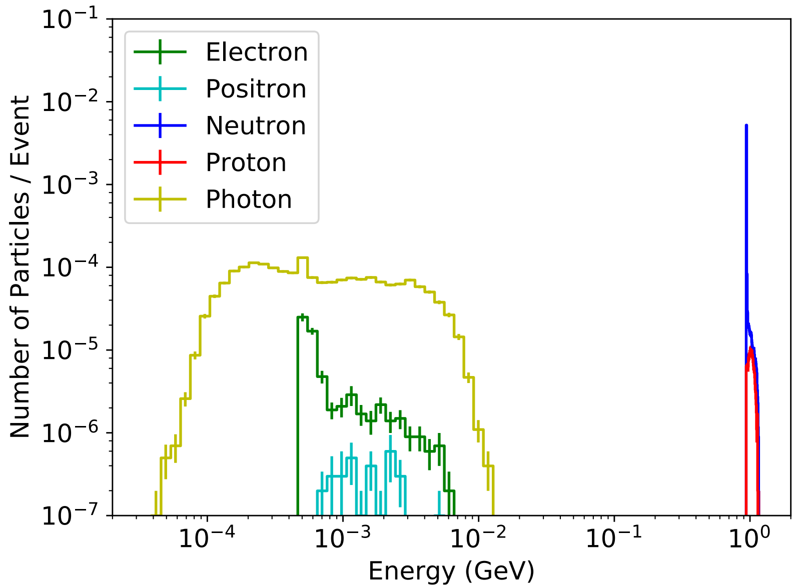
Horizontal phase space before (left) and after (right) a degrader.



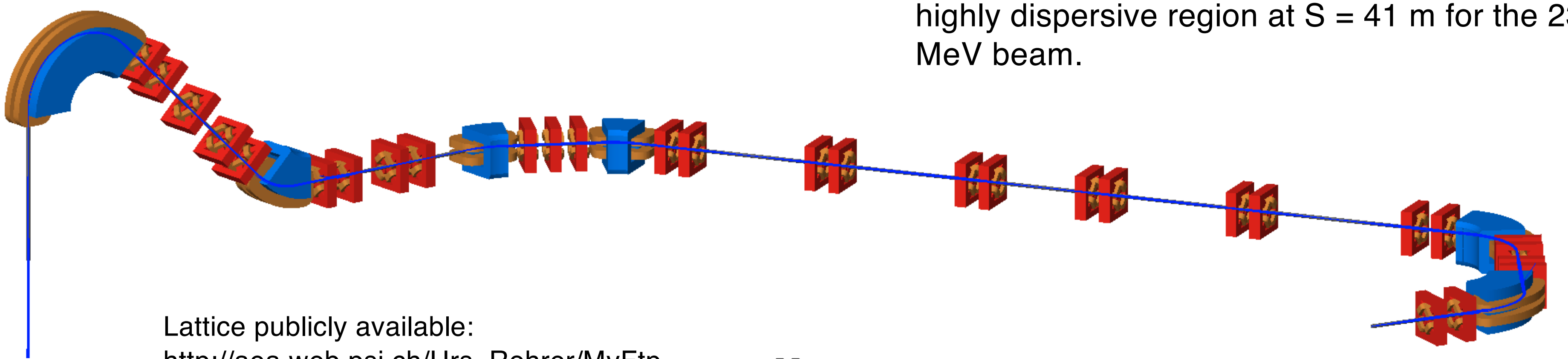
PSI Gantry II

- Optical comparison and validation





Secondaries generated from primary losses in a highly dispersive region at S = 41 m for the 230 MeV beam.



Lattice publicly available:
http://aea.web.psi.ch/Urs_Rohrer/MyFtp

Published at IPAC 2018 - MOPML061

Quality & Testing

- Open source C++ software in git repository
  - https://bitbucket.org/jairhul/bdsim/wiki/Home
- Nightly testing of ~ 600 tests
  - 6 builds, SLC6 & CC7
  - > 90% code coverage
  - regression testing
- Public issue tracker
  - https://bitbucket.org/jairhul/bdsim/issues
  - also for feature requests
- Complete Doxygen documentation for C++
  - http://www.pp.rhul.ac.uk/bdsim/doxygen/
- Detailed manual regularly updated
  - http://www.pp.rhul.ac.uk/bdsim/manual/
  - html & pdf

Getting Started & Setup

- Build with complete environment available on AFS at CERN:
  - source /afs/cern.ch/user/l/lnevay/work/public/bdsim-1.2.1.sh
- Local install requires:
  - Geant4 (4.10.2.p02 recommended)
    - recommended with GMDL and Qt visualisers
  - ROOT 6
  - CLHEP
  - Bison
  - Flex
  - Xercesc3 for GDML
  - QT for nicest visualiser

Development & Collaboration

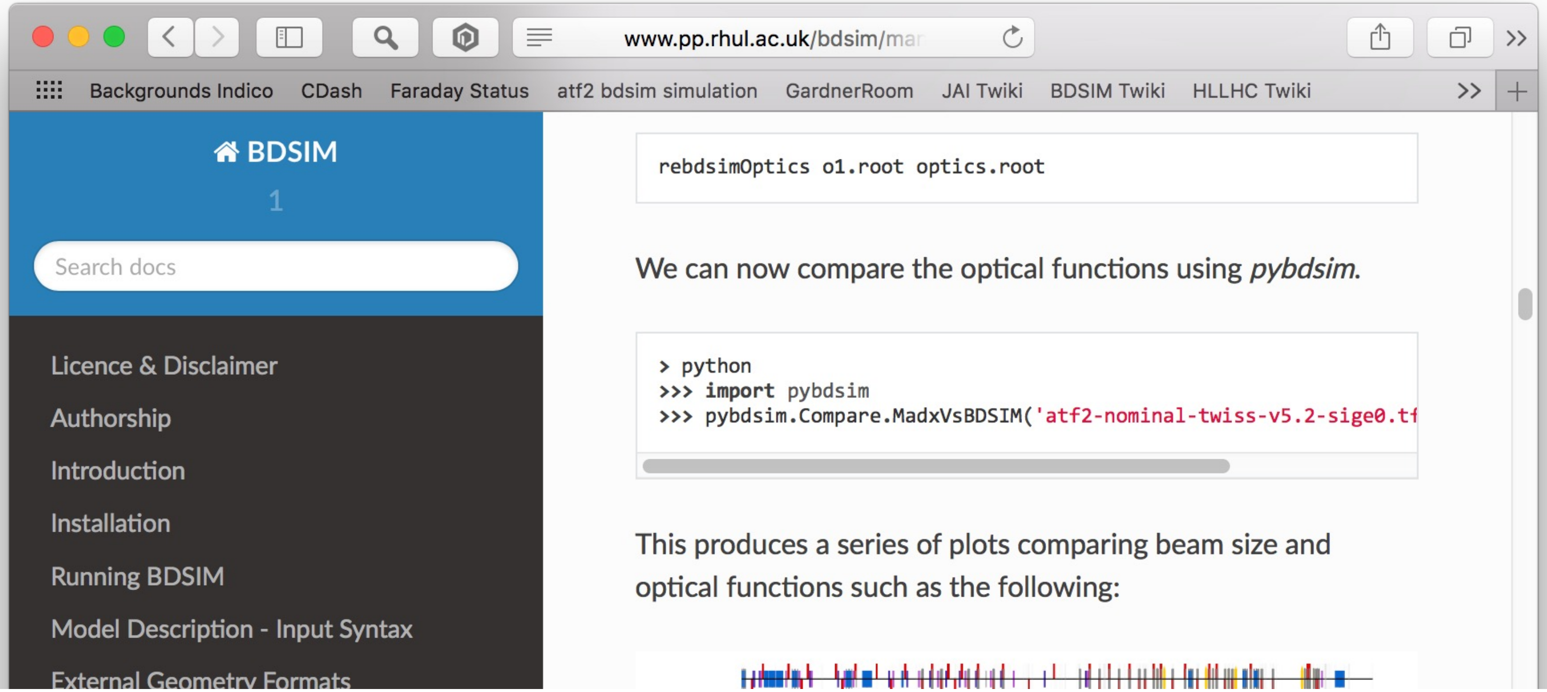
- BDSIM highly suited for many studies now
- However, open source and constantly being developed
- Collaboration welcome
- Please discuss with us!
- Email laurie.nevay@cern.ch

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Thank you

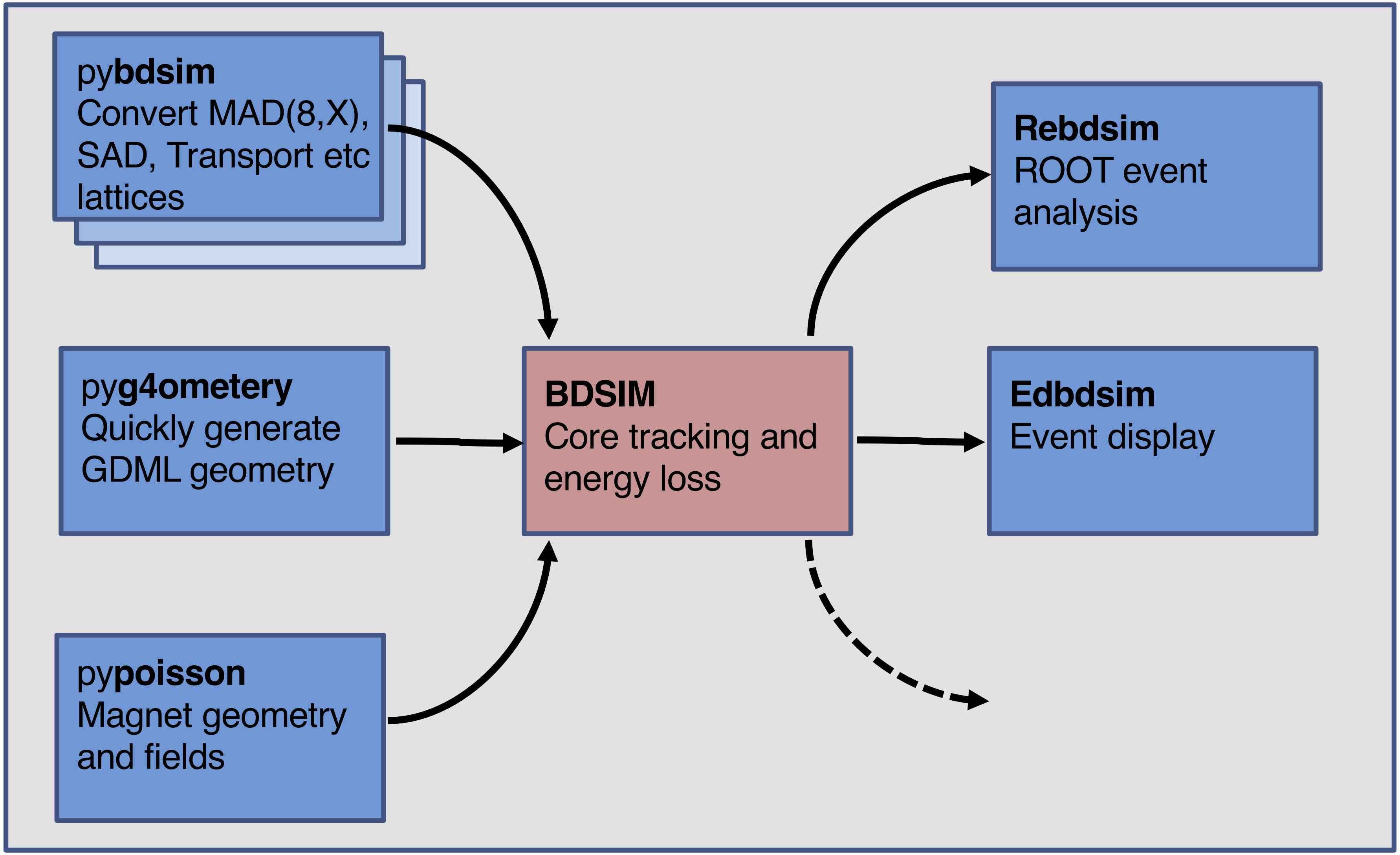
Links

- paper: *https://arxiv.org/abs/1808.10745*
- main website: http://www.pp.rhul.ac.uk/bdsim
- manual: http://www.pp.rhul.ac.uk/bdsim/manual
- git repository: https://bitbucket.org/jairhul/bdsim/wiki/Home
- Issue tracking & feature request
  - https://bitbucket.org/jairhul/bdsim/issues



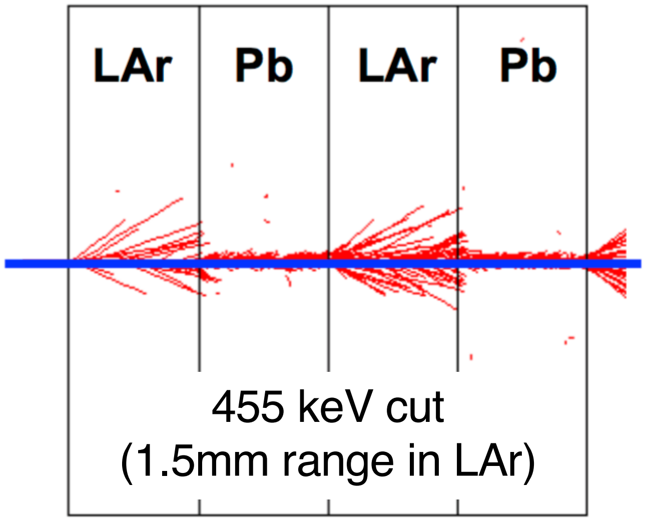
backup

BDSIM ecosystem

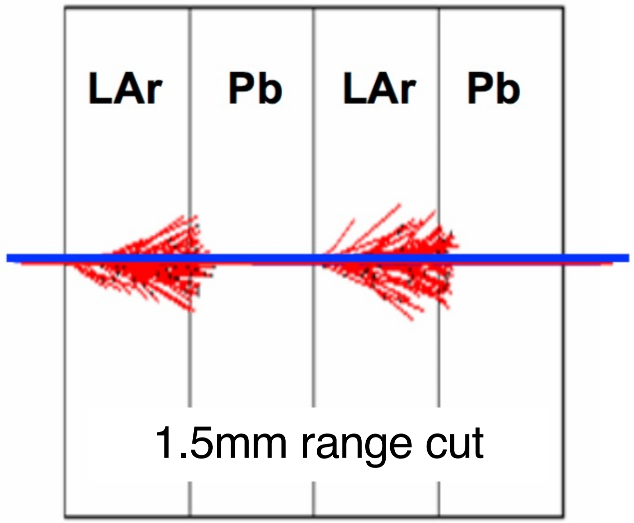


Control Over Simulation

- Huge number of secondaries
  - e.g. 10^4 secondaries / event
- 'Infrared divergence'
- Necessary but can dominate tracking time
- Control through production 'range' cuts
- Roughly distance secondary would have to travel
  - corresponds to a different energy / particle / material



Geant4 example:
500 MeV p in
Lead sampling
calorimeter



CLIC Post Collision Line