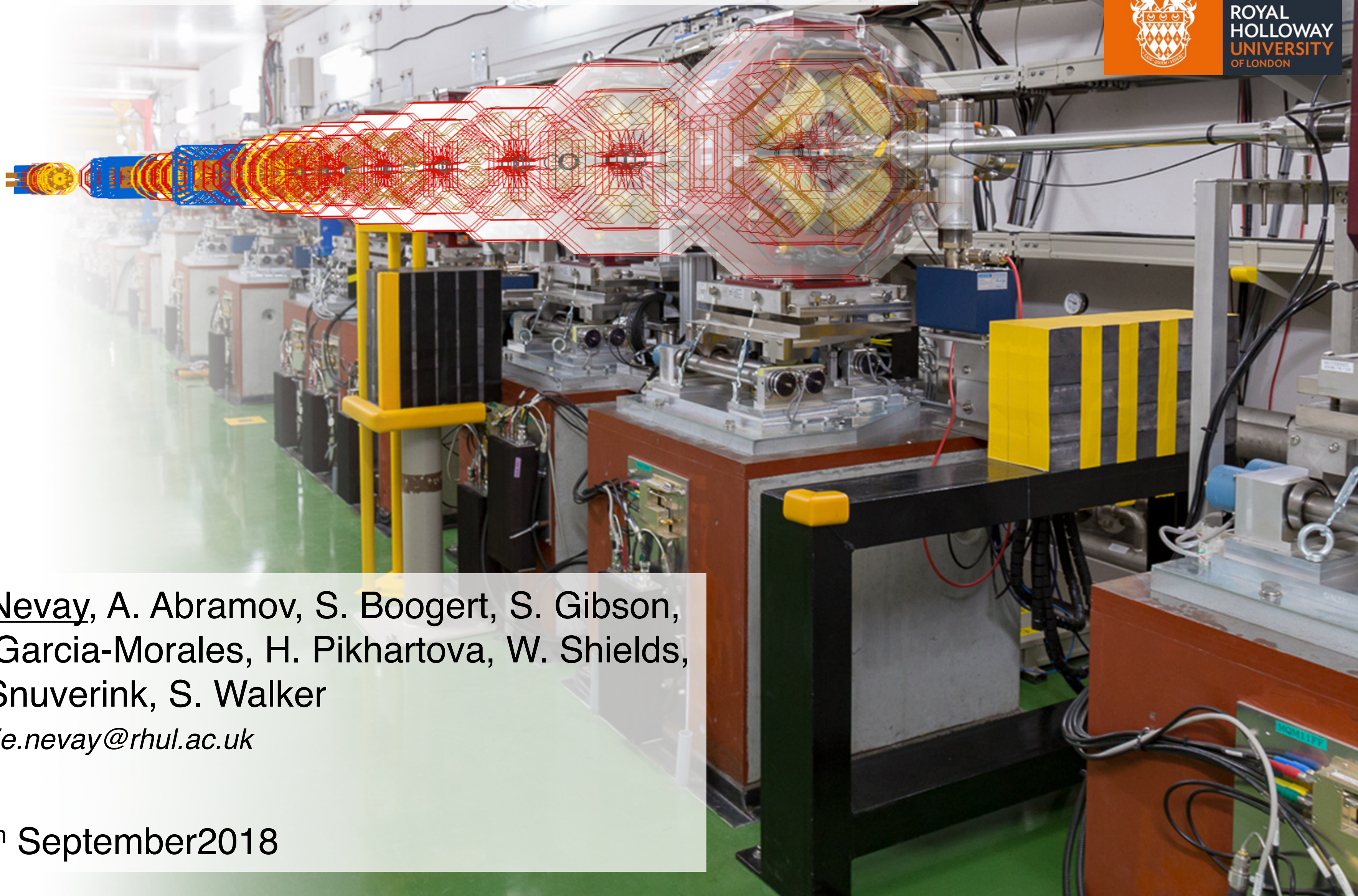
Mixed Accelerator and Particle Physics Simulations



John Adams Institute for Accelerator Science

Royal Holloway University of London

<u>L. Nevay</u>, A. Abramov, S. Boogert, S. Gibson, H. Garcia-Morales, H. Pikhartova, W. Shields, J. Snuverink, S. Walker

laurie.nevay@rhul.ac.uk

24th September2018

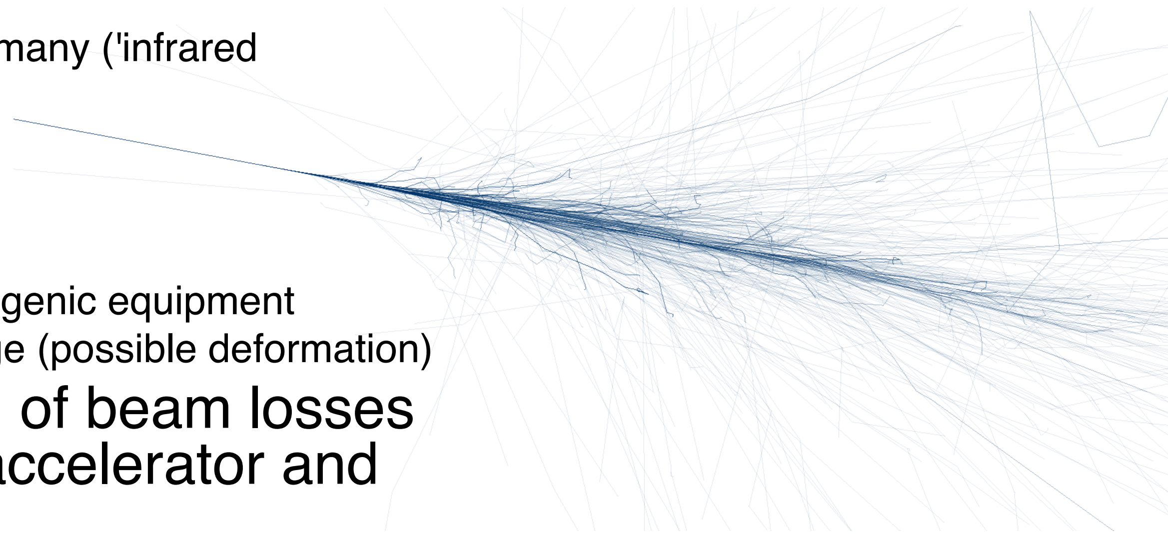
Outline

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

Introduction

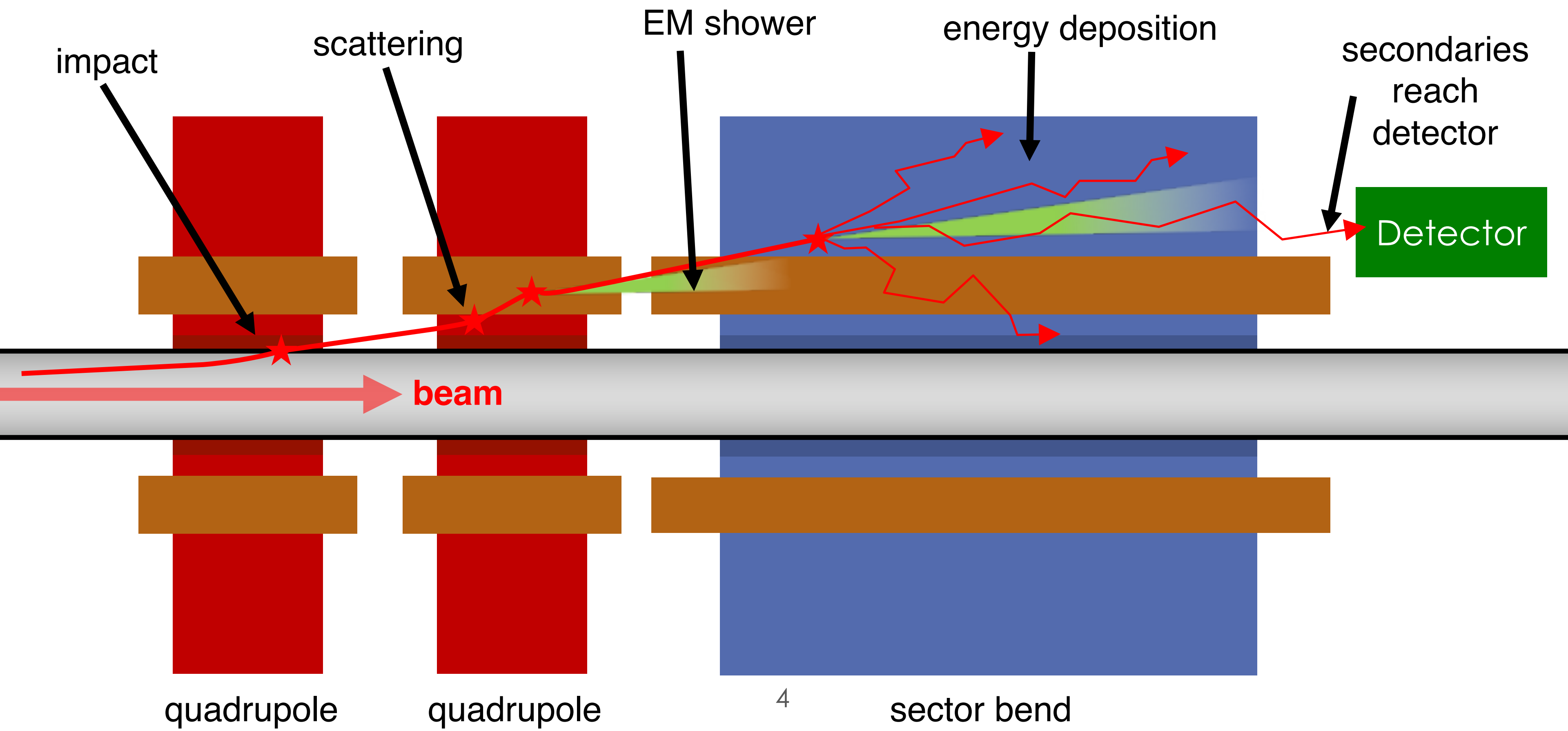
- No accelerator perfectly contains all particles
  - either by design tolerance (certain capture %)
  - via stochastic processes (intra-beam scattering, beam-beam, collective effects)
- High energy particles lead to many secondary particles
  - photons, e^-, e^+, nuclear fragments, exotic unstable particles, etc.
  - one initial particle leads to many ('infrared divergence')
- Beam loss leads to:
  - detector background
  - energy deposition
  - heat loads, possibly in cryogenic equipment
  - radio-activation and damage (possible deformation)
- Prediction and control of beam losses is crucial to both the accelerator and experiment operation

1x 10 GeV e^- in copper



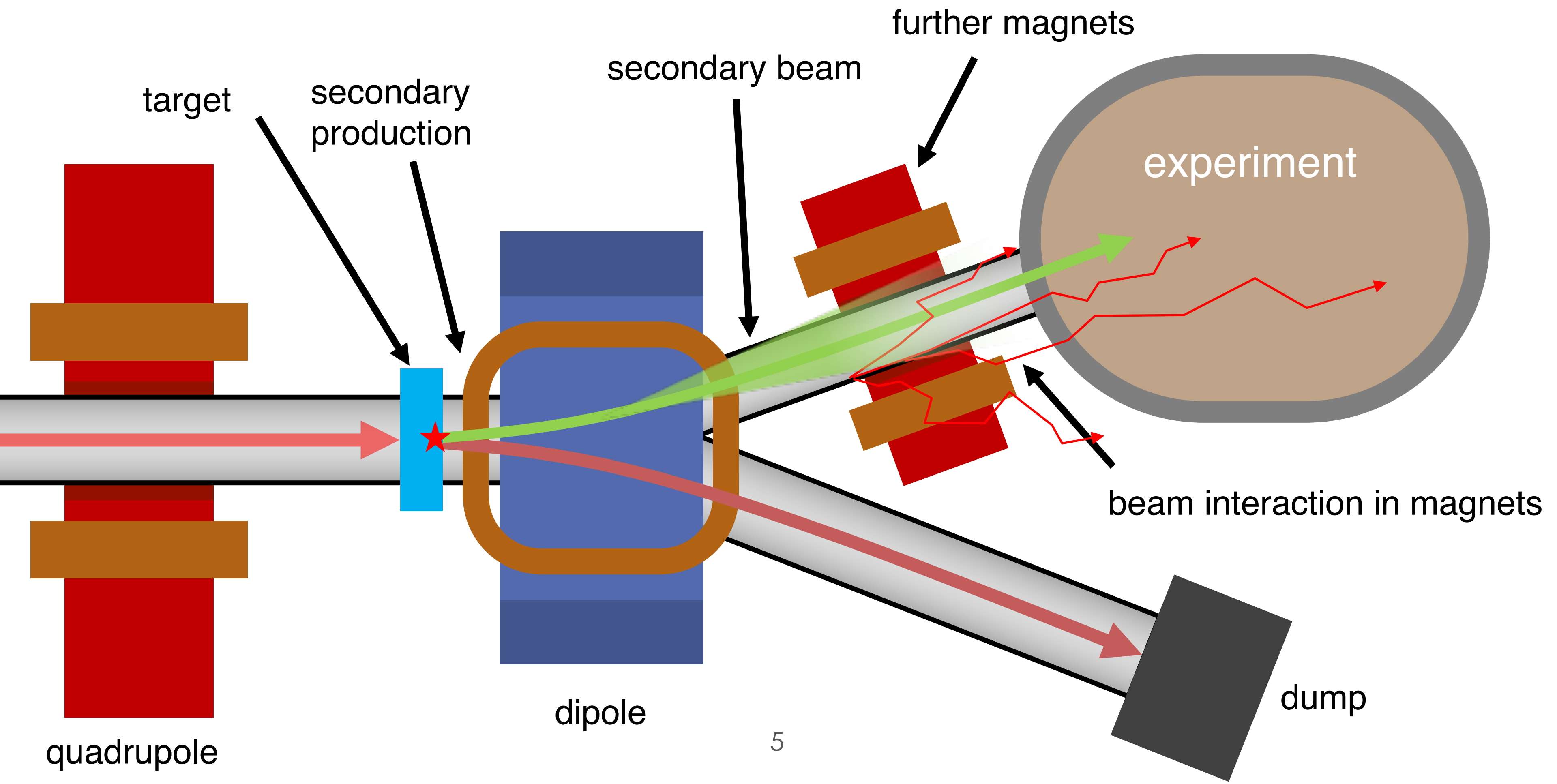
Beam Loss

- Cut-through of accelerator
- Particle impacts aperture at some point
- Secondary particles and radiation propagate some distance
- Energy deposited in many components



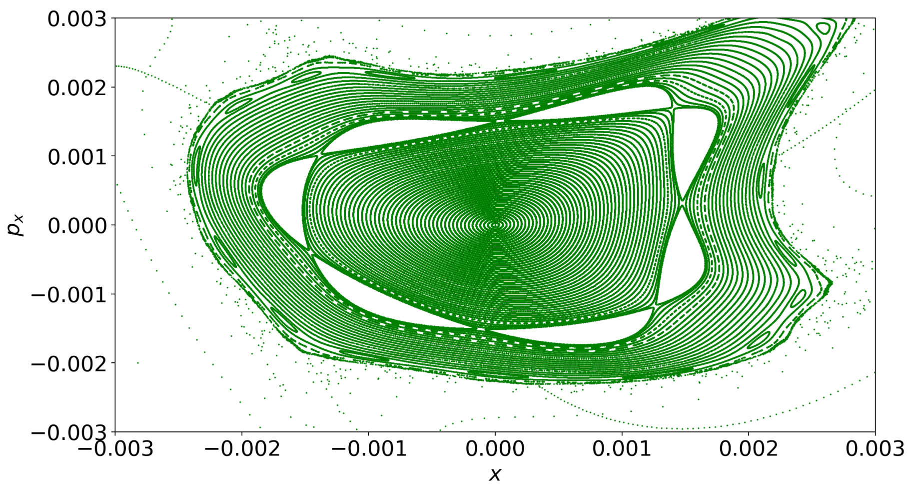
Secondary Beam

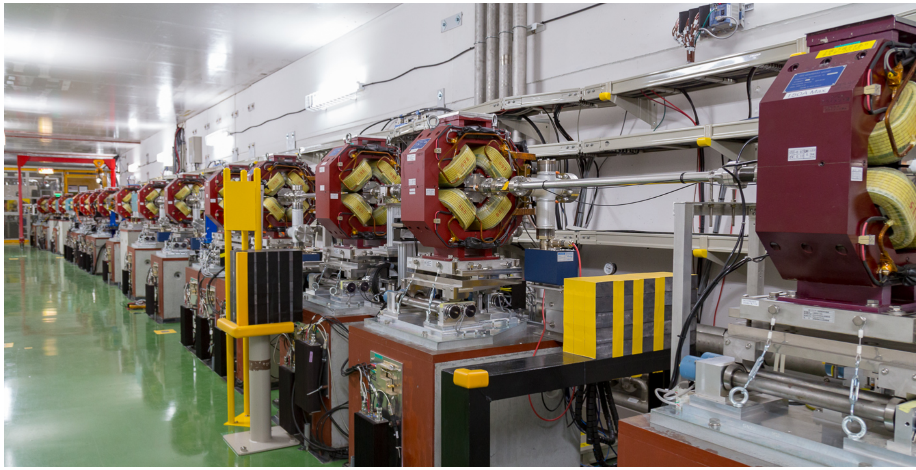
- Secondary particle production from impact with target
- All of beam impacts target
- Both beams transported in magnets afterwards



Accelerator Tracking

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

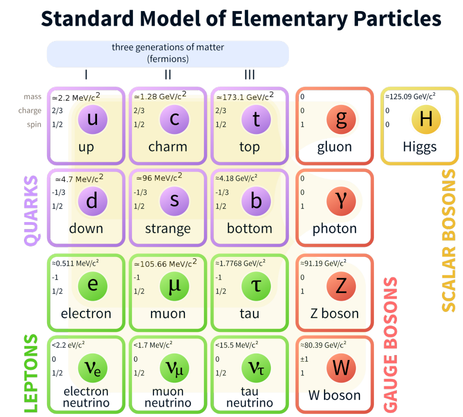


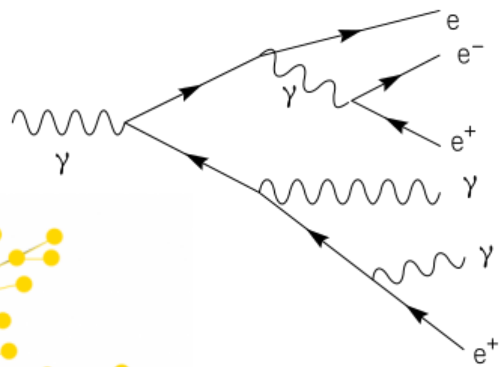


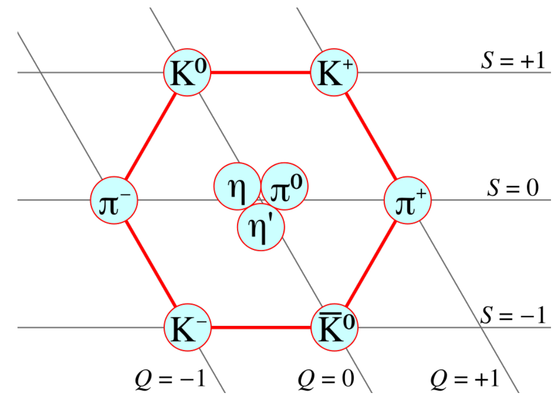
Example Poincaré map through nonlinear fields

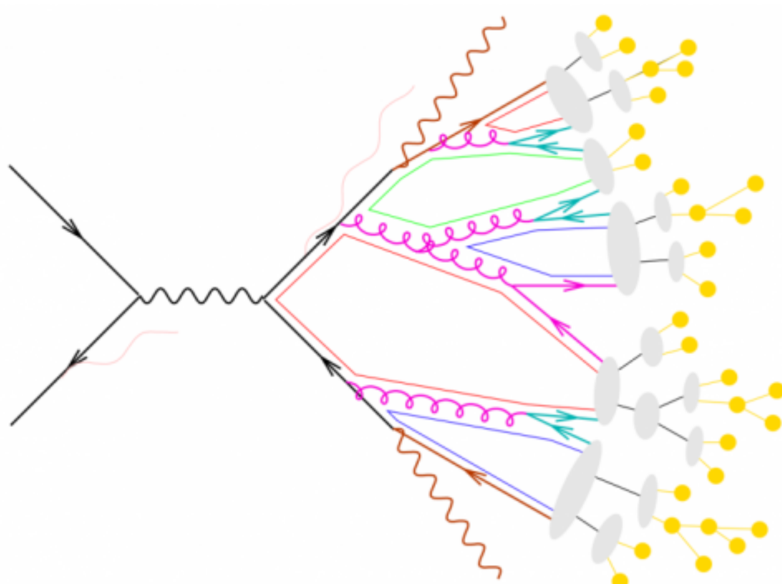
Particle Physics Processes

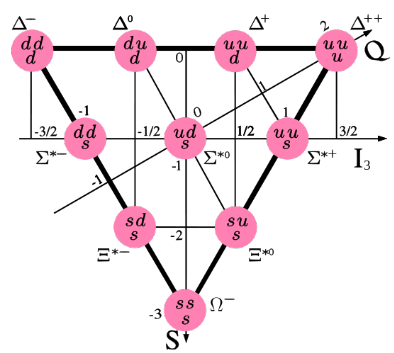
- Large variety of particles
- Large variety of processes & models
  - some data based, some pure model based, some mixed
  - different models for different energy ranges
- Available libraries - FLUKA, Geant4, MARS











Existing Solutions

- 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

- SixTrack
- PTC / MADX
- Transport
- Lucretia

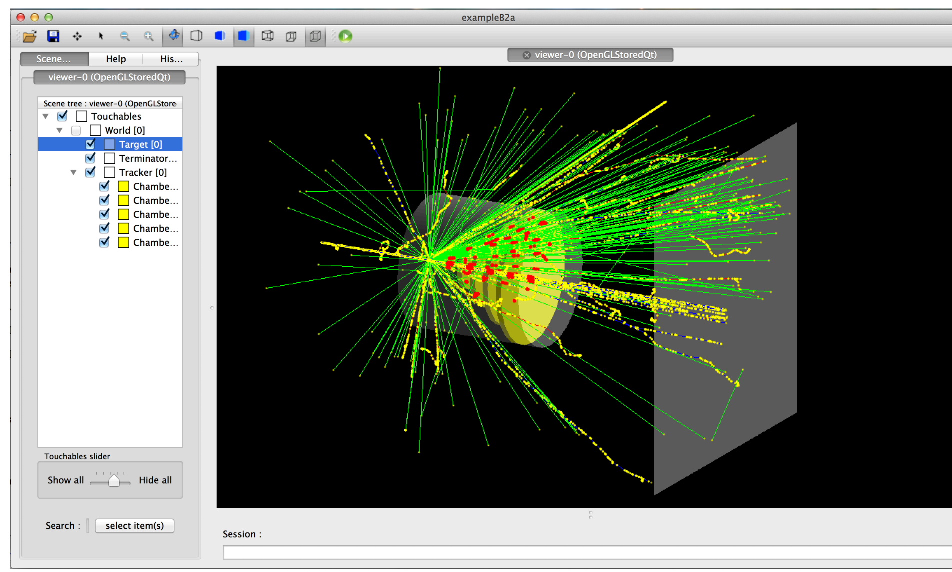
Radiation Transport

- FLUKA
- Geant4
- MARS
- 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

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

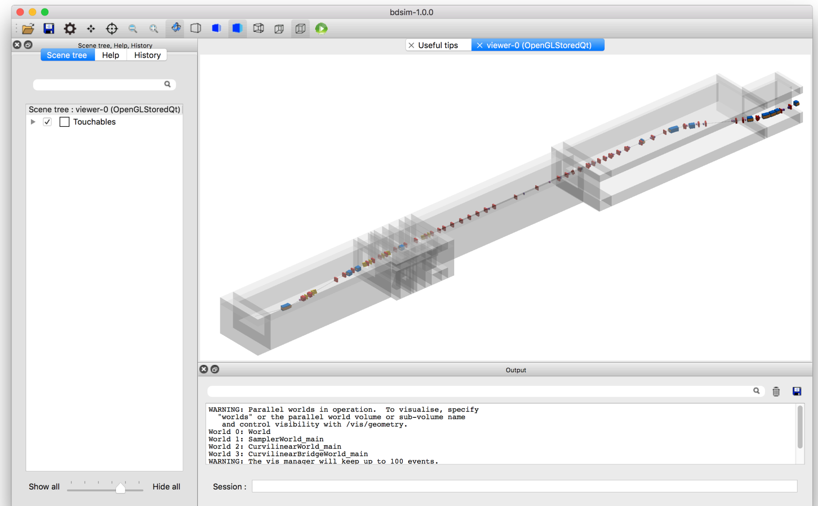
- Tracking codes complex in implementation
- Speciality can vary depending on application

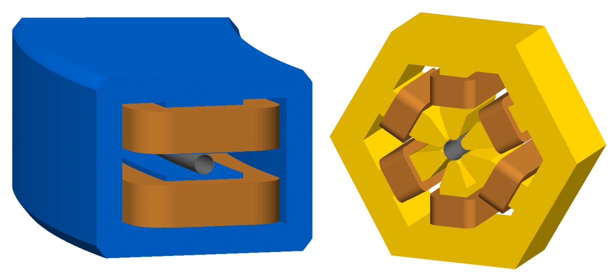
- rarely do people therefore make such a model…

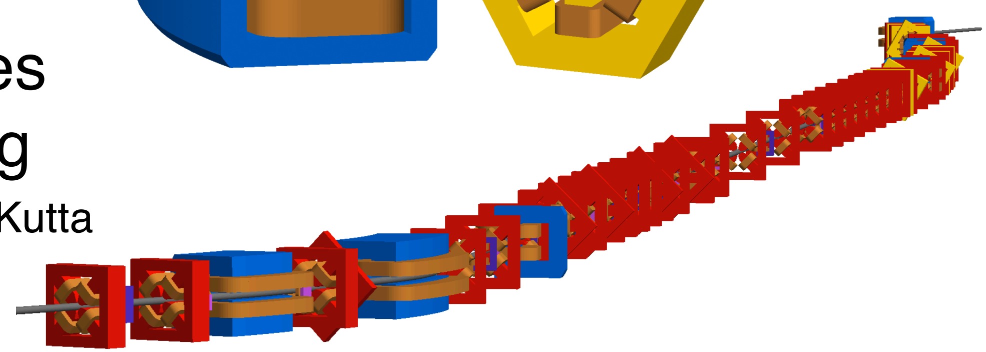
Beam Delivery Simulation

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 1st order matrices used for in-vacuum tracking
  - replaces Geant4's 4th order Runge-Kutta







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 / Transport / Sixtrack
  - only particle tracking -> no matrix propagation
- Prepare model from optical description

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

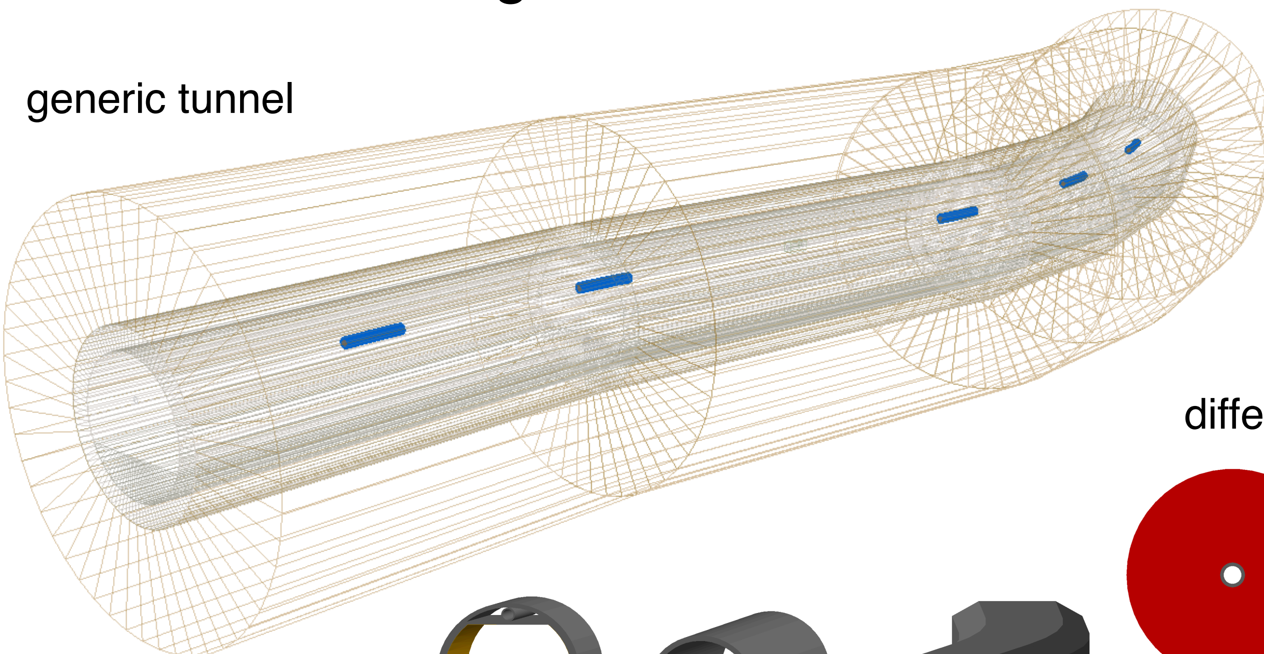
```
>>> drift123dict = {'aper1':0.03, 'aper2':0.05, 'apertureType':'rectangular'}
>>> quaddict = {'magnetGeometryType':'polesfacetcrop}
>>> d = {'drift123':drift123dict, 'qf1x':quaddict}
>>> a,o = pybdsim.Convert.MadxTfs2Gmad('inputfile.tfs', 'latticev1', userdict=d)
```

Geant4 Model Ingredients

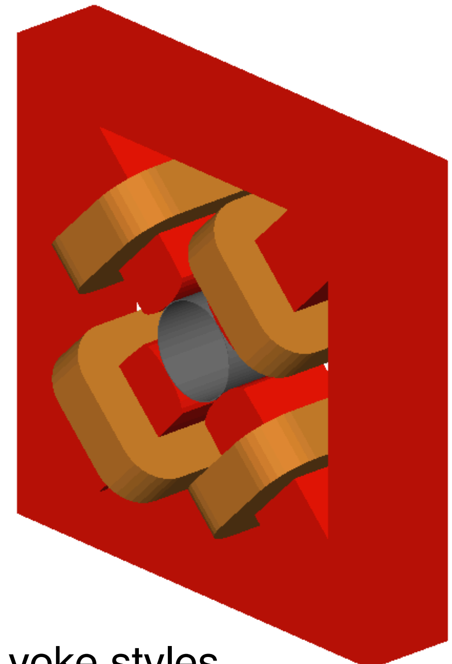
- Requires definition of
  - geometry
  - fields
  - physics processes
- Library of scalable generic geometry provided
- Matching perfect fields for each magnet provided
  - ideal multipole for yoke
- Simple interface to Geant4's modular physics lists and reference physics lists
  - modular -> "em", "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)

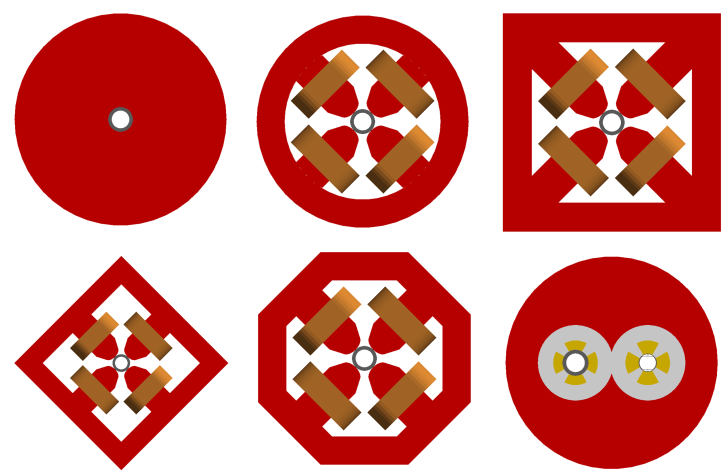
Generic Geometry

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

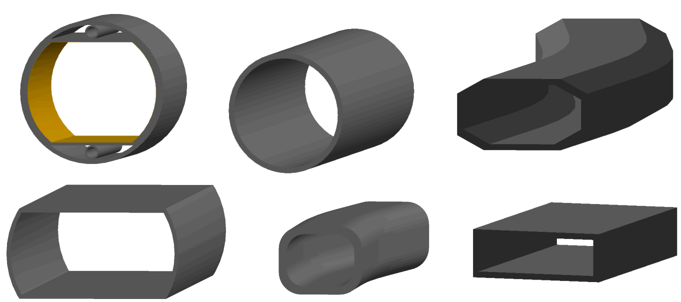


generic tunnel





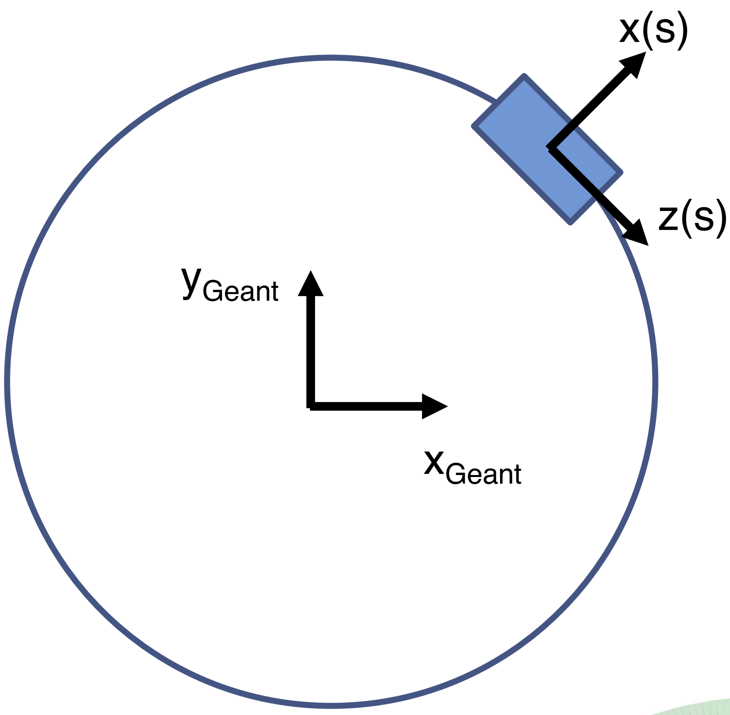
different yoke styles

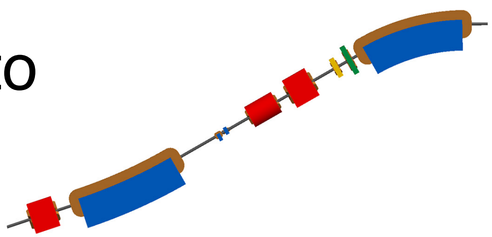


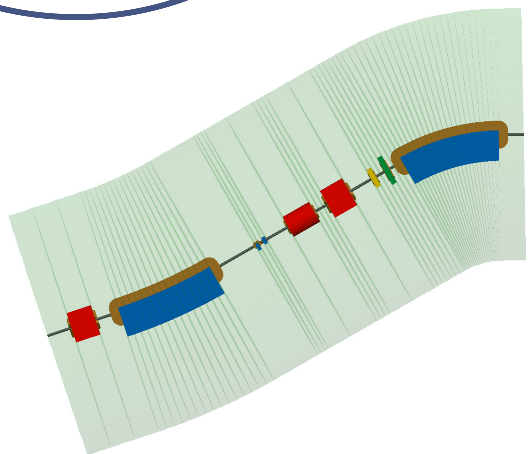
all aperture types from MADX available

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



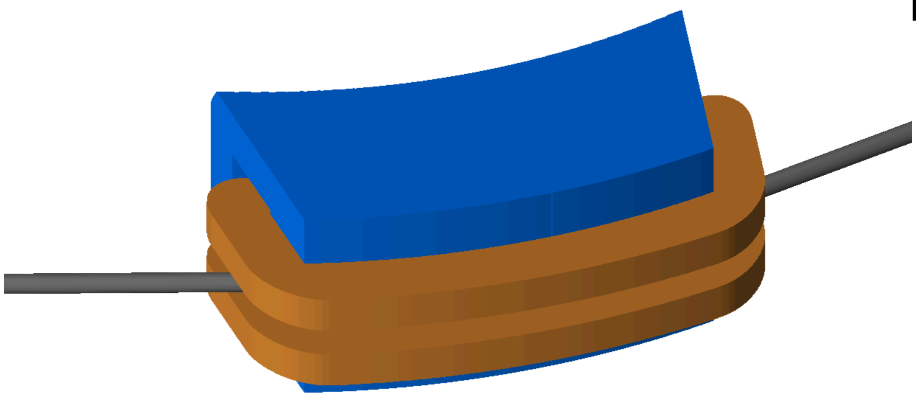




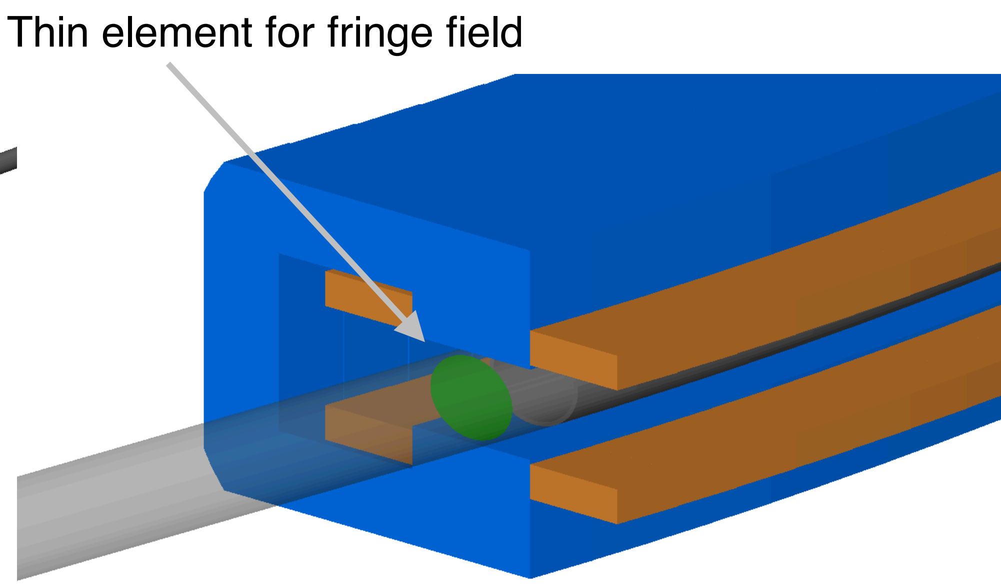
Pole Faces & Thin Elements

- 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



Physics Processes

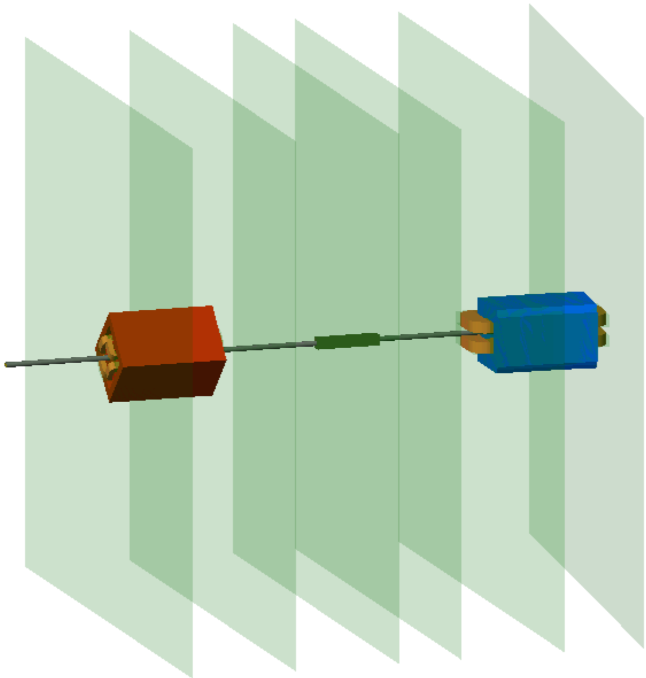
- Huge number of physics processes in Geant4
  - No one model for all particles at all energies
- Use modular physics 'lists'
  - standard set of processes for application / energy range
- "G4EmStandardPhysics" for example -> electromagnetic
- hadronic, decay, muon-specific, synchrotron radiation etc.
- Only use physics required
  - more processes = slower simulation
  - only selection of physics processes relevant for any application
  - possibility of different models for different energy ranges

```
option, physicsList = "em ftfp_bert decay muon hadronic_elastic em_extra"
```

Information Reduction

- A particle physics simulation produces a potentially huge 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 particles
- 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

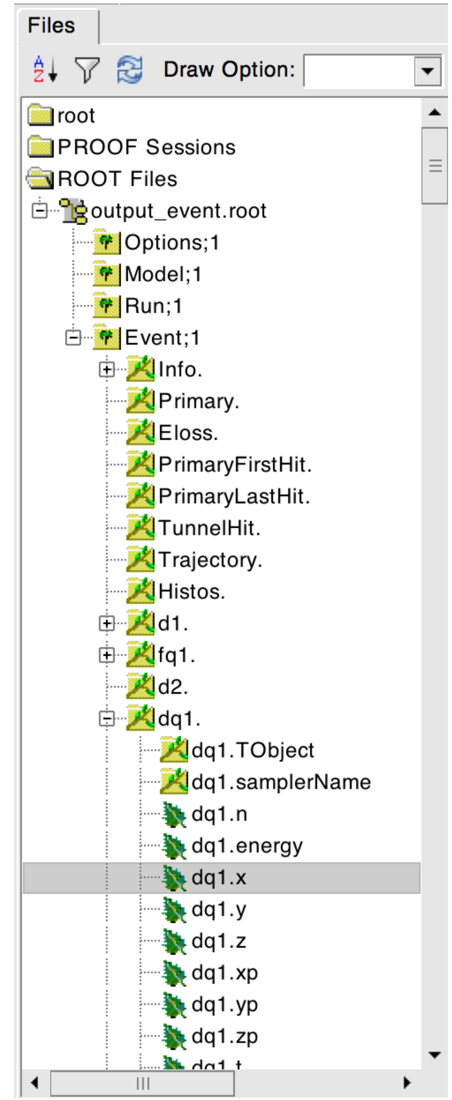
sampling planes after each element (normally invisible)

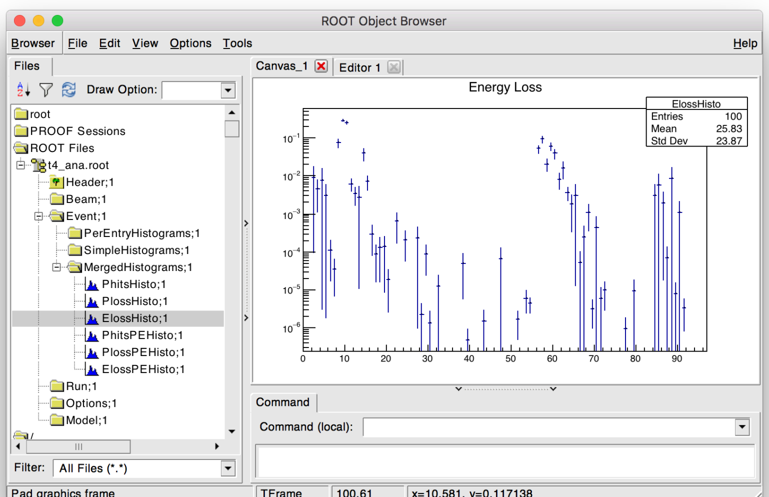


Output

- 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 very large data sets
- Specifically designed for data evolution
- Strong reproducibility
  - all random number generator seeds and settings stored

structure of an output file



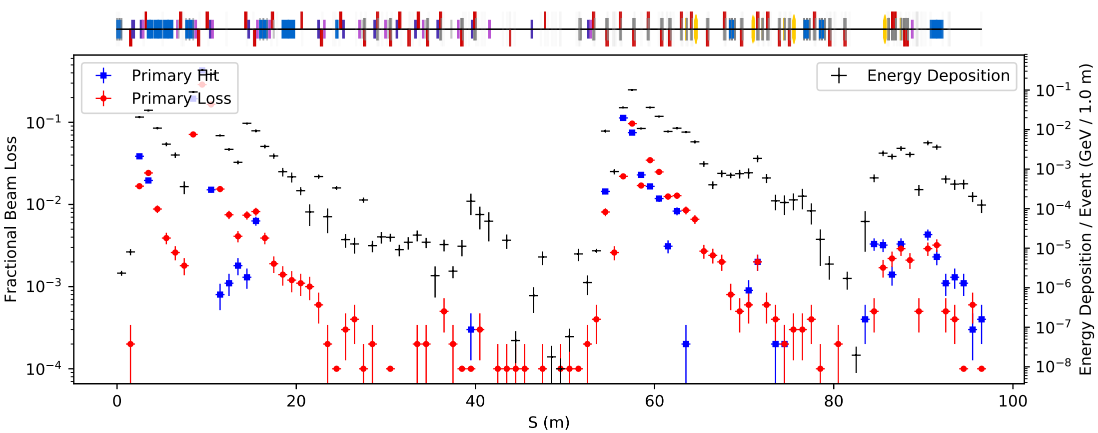


Analysis

- 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)
```

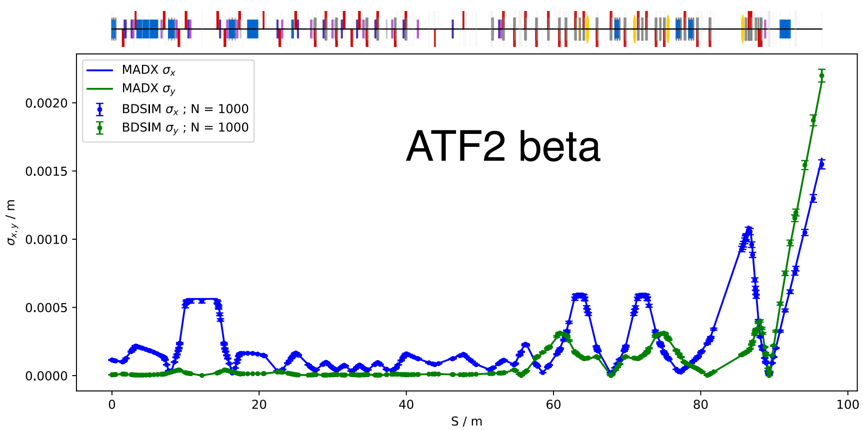


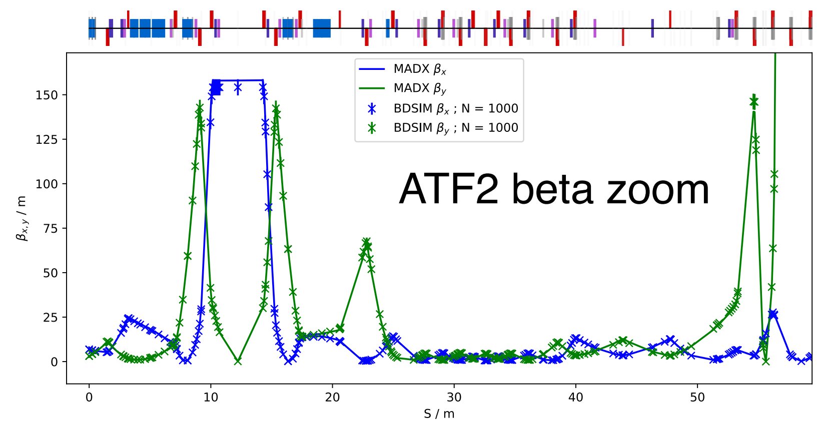
'selection' is weighting can be Boolean and or numerical

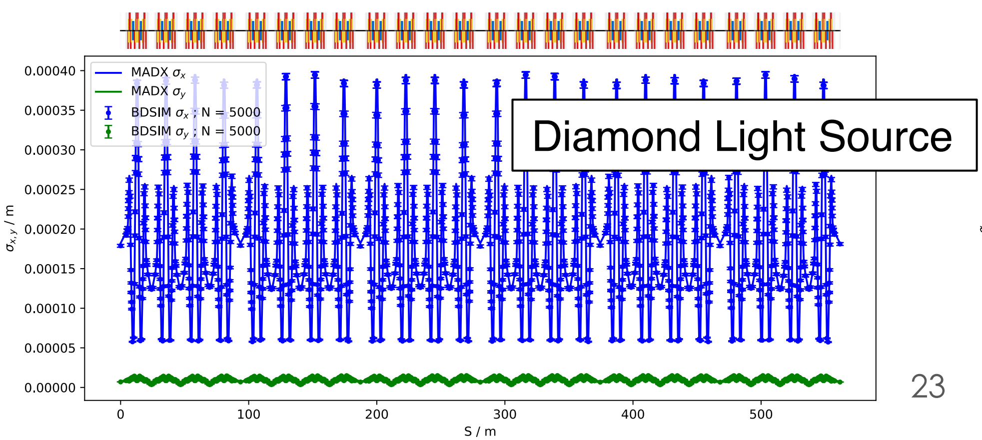
basic impact, loss and energy deposition analysis

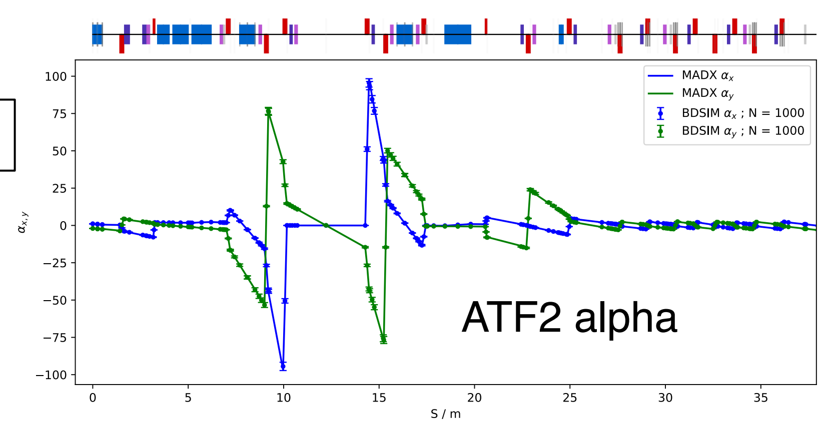
Optical Function Comparison

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





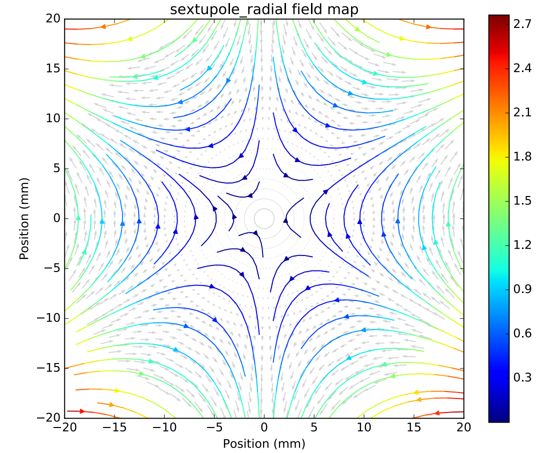




Adding more detail…

Field Maps

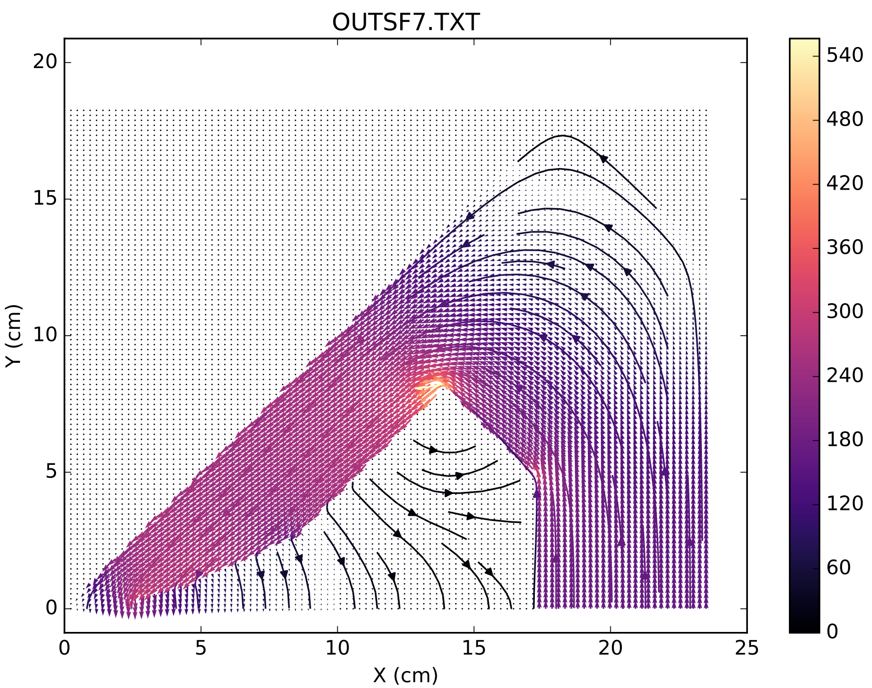
- Equations describe pure fields
- Can overlay field map on BDSIM generic element
  - yoke or vacuum separately or both together
- 1- 4D loading and interpolation
  - nearest neighbour, linear and cubic interpolation



ideal sextupole field

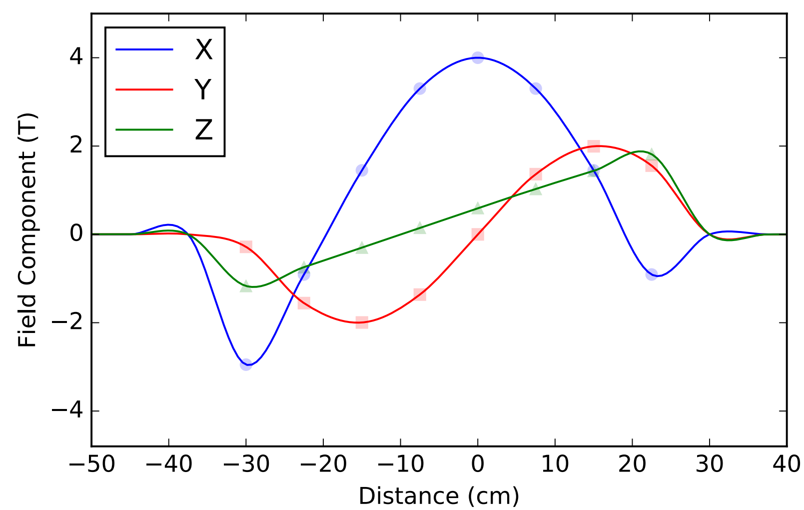
$$B_x = \frac{1}{2!}\frac{\partial^2 B_y}{\partial x^2}\, 2xy$$

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



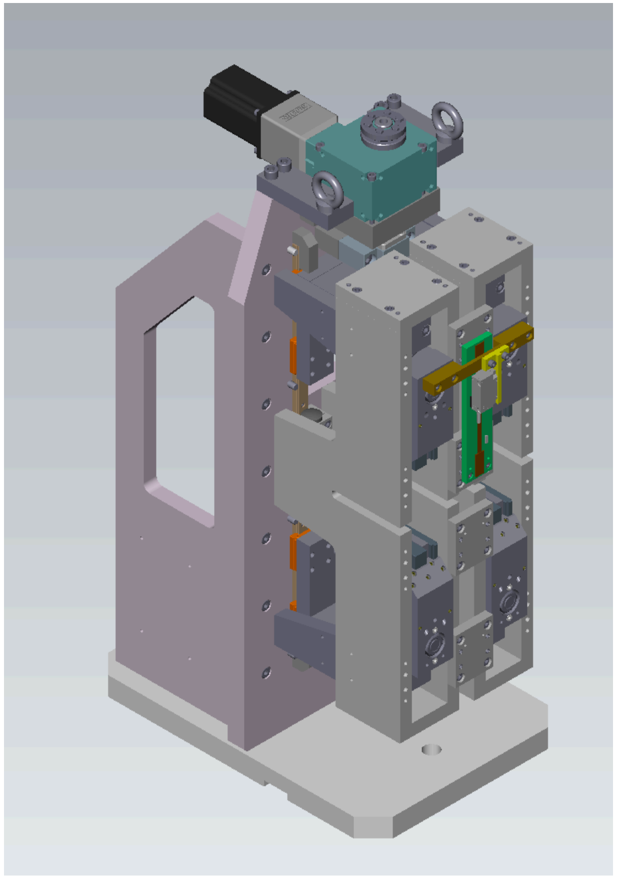
example field map from Poisson 2D

example interpolation



Externally Provided Geometry

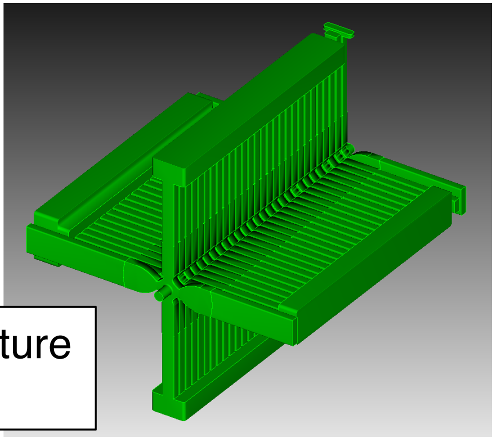
- Most devices designed in CAD
- Common desire to use CAD model for radiation studies
- Pieces can be converted:
  - individual STL (water bag mesh) per component
  - STEP file (more structure)
- Often these are too complex
  - bolt holes, screws
  - pieces grouped by material rather than location
- Technically possible to convert but often inefficient for final simulation
- Must choose level of required detail and how important it is



Example DESY phase shifter with actuators

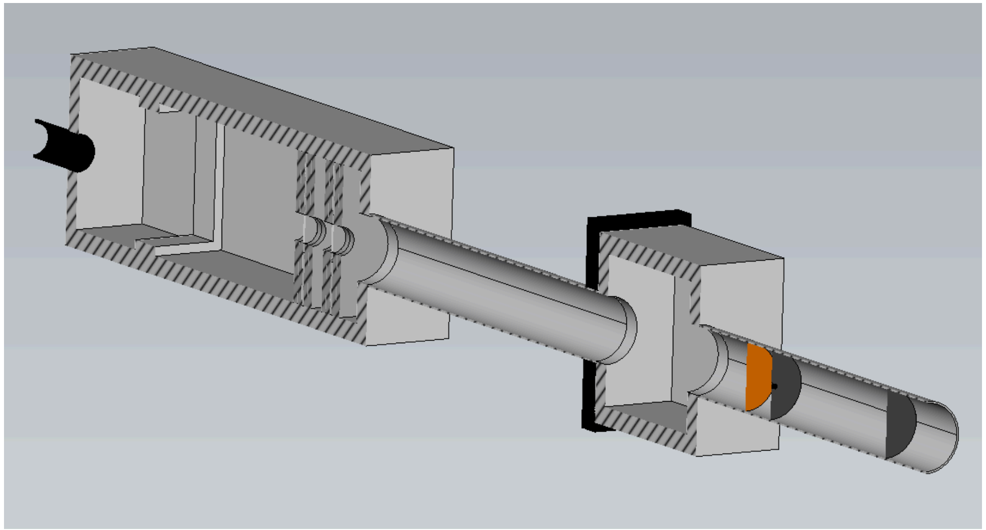
Complex Models

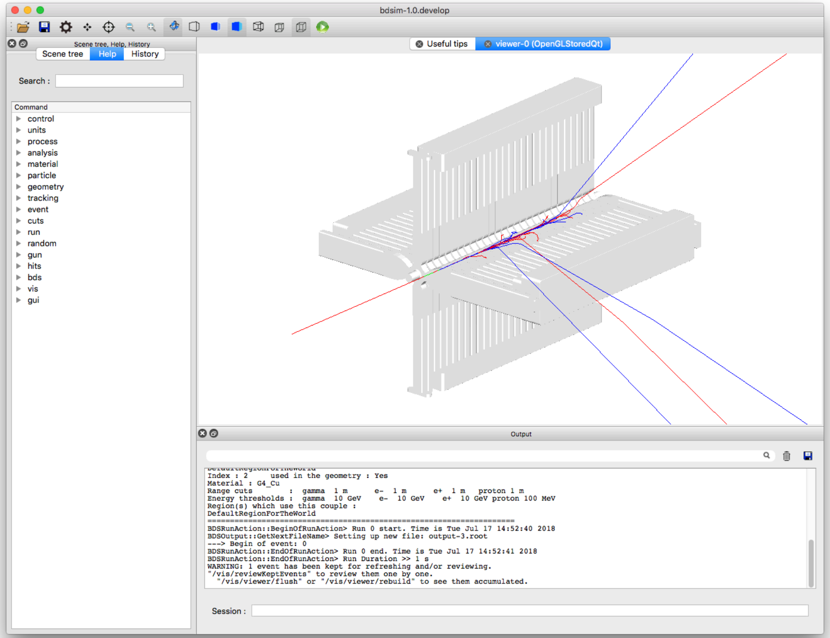
- Developed Python package to process CAD models
  - "pyg4ometry"
- Create mesh from STEP file
  - using Open Cascade and FreeCAD (free) tools
- Smaller models more suited



CLIC vacuum structure from STL file

Clatterbridge occular treatment nozzle



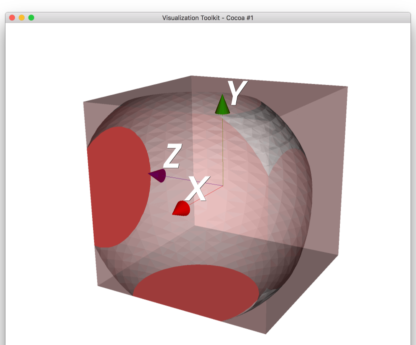


pyg4ometry

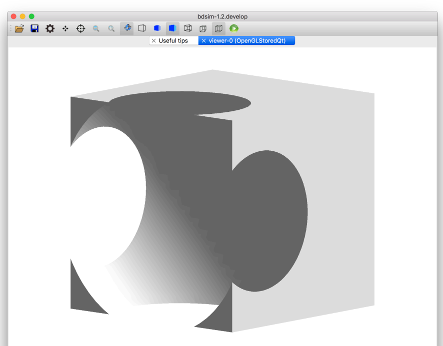
- Python package to create Geant4 geometry
- Python class for each Geant4 primitive solid
- Combine with meshes from STL / STEP
- Exports to GDML format for use in Geant4 / BDSIM

```
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)
```

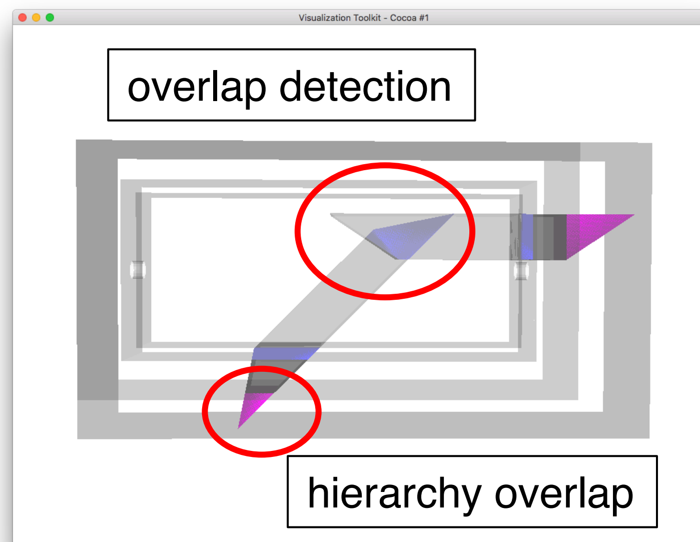
- Creates its own mesh
- Use mesh to identify solids
- Easy to create simplified pieces



VTK visualiser

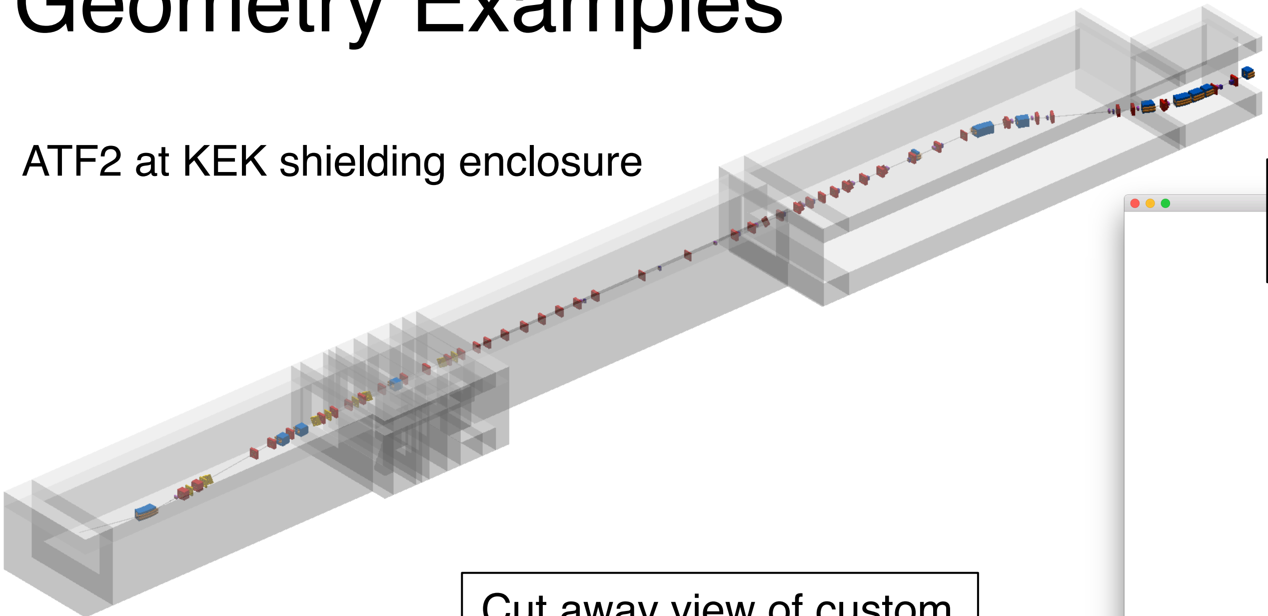


BDSIM

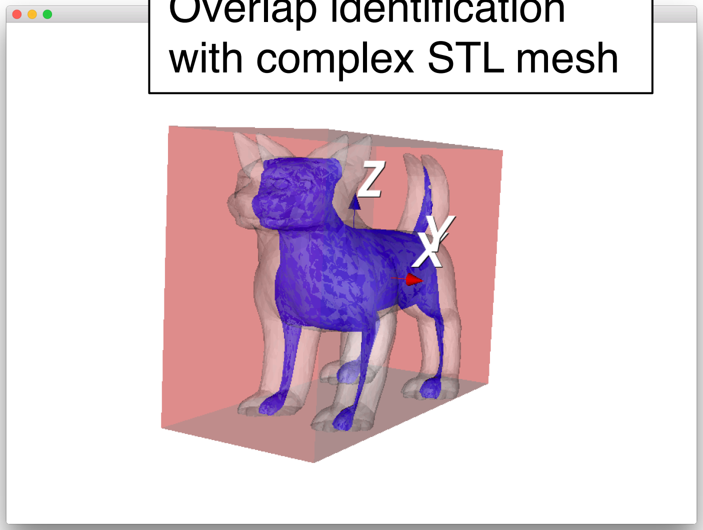


Geometry Examples

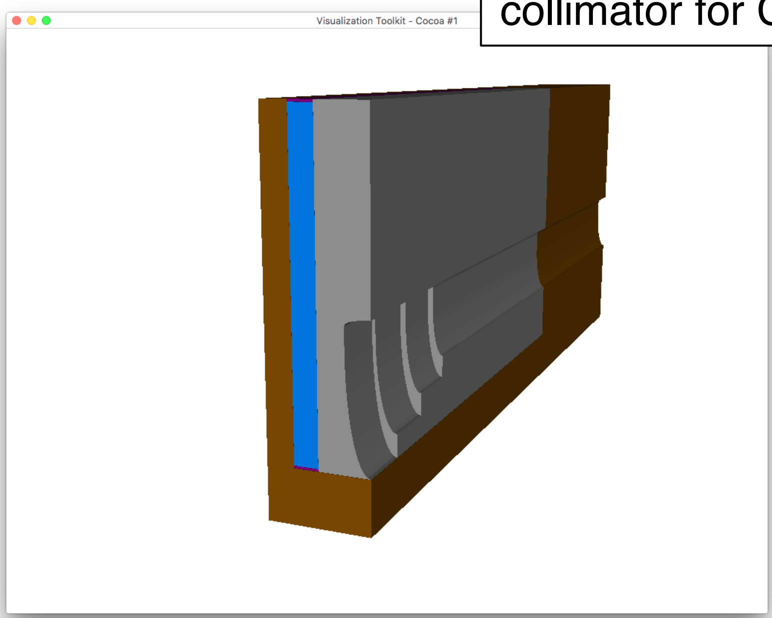
ATF2 at KEK shielding enclosure

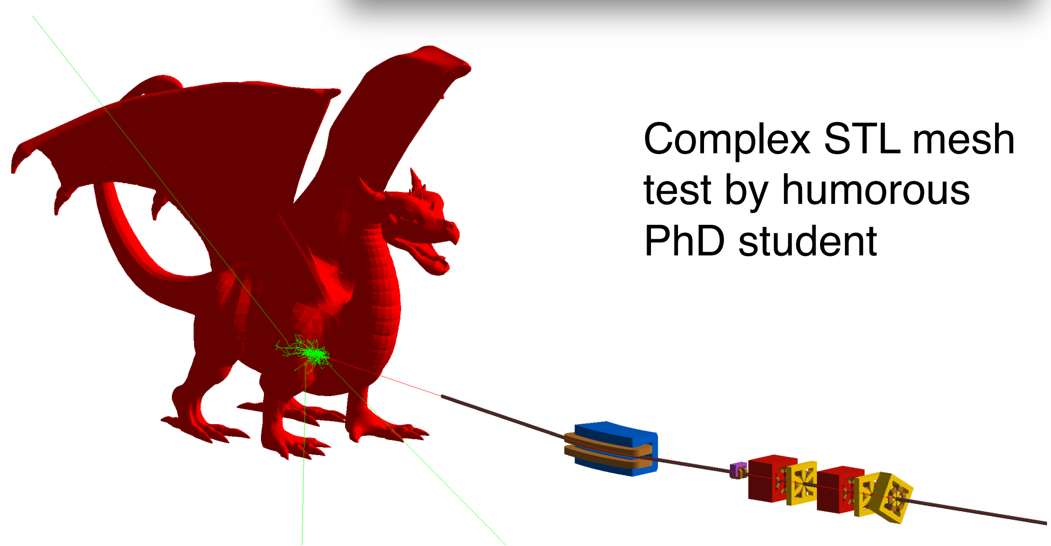


Overlap identification with complex STL mesh



Cut away view of custom collimator for CLIC



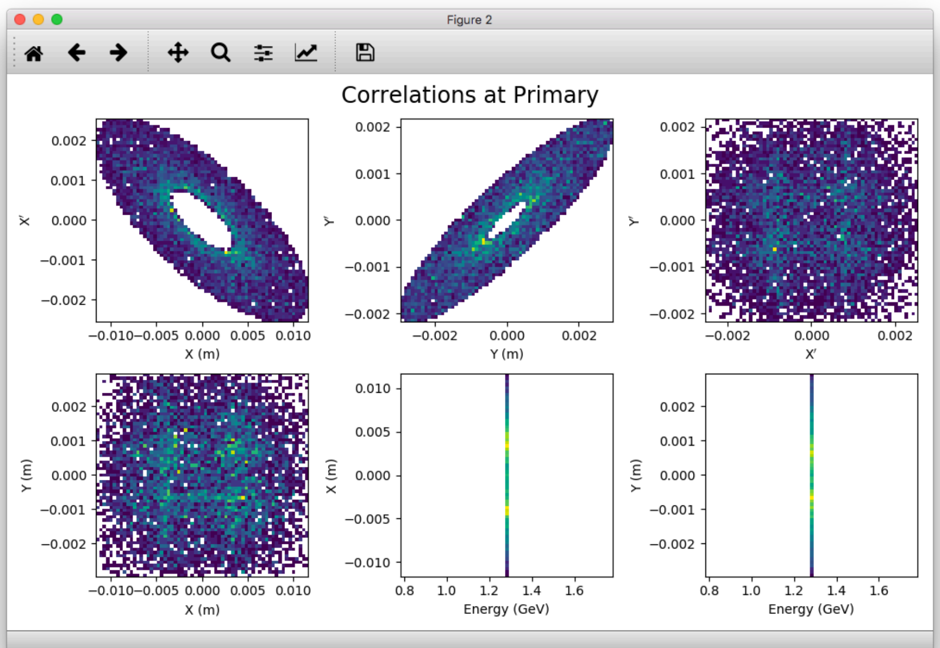


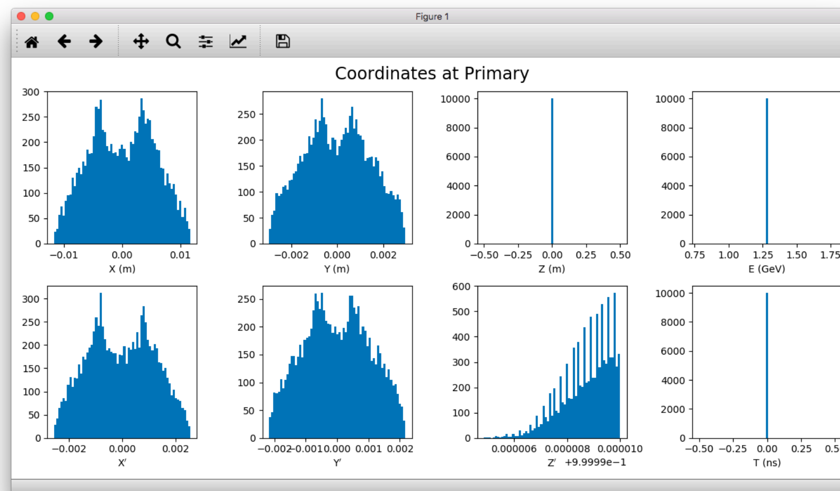
Complex STL mesh test by humorous PhD student

Control & Efficiency

Beam Distribution

- Beam interaction and loss can be rare
- Interaction at 50 sigma?
- Efficiently generate required distribution
- Import beam distribution from ASCII
  - compressed ASCII also accepted

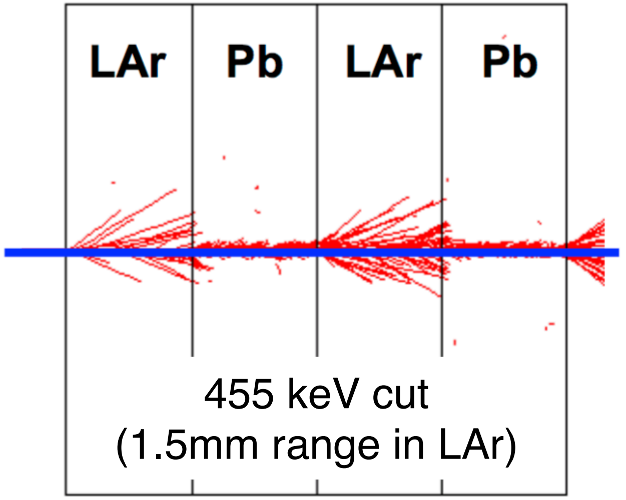




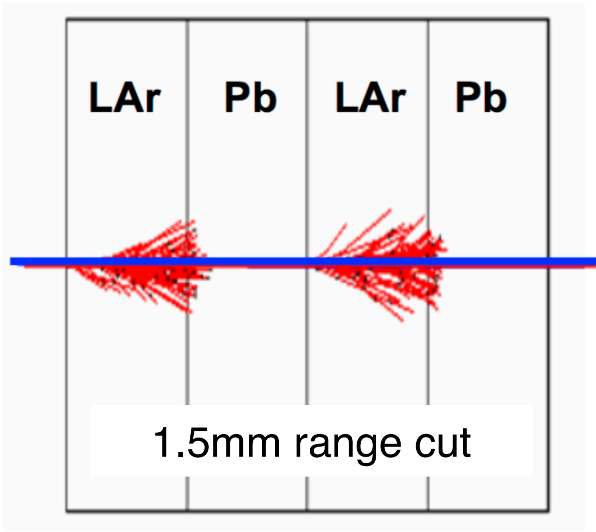
variety of distributions included using CLHEP pseudo-random number generator

Secondary Particles

- Huge number of secondaries
  - e.g. 10^4 secondaries / event -> often 10^2 to 10^7 events simulated
- '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

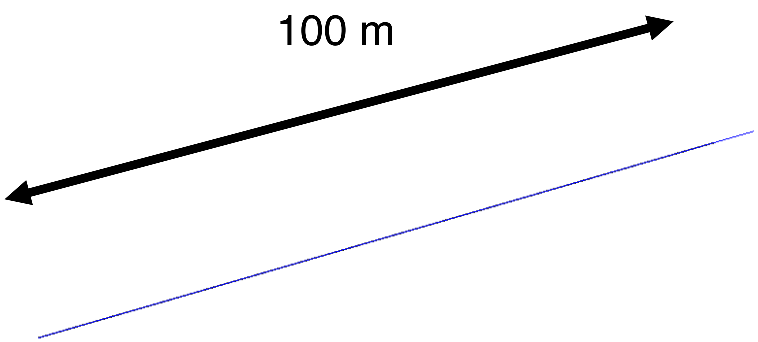


Geant4 example:
500 MeV p in
Lead sampling
calorimeter

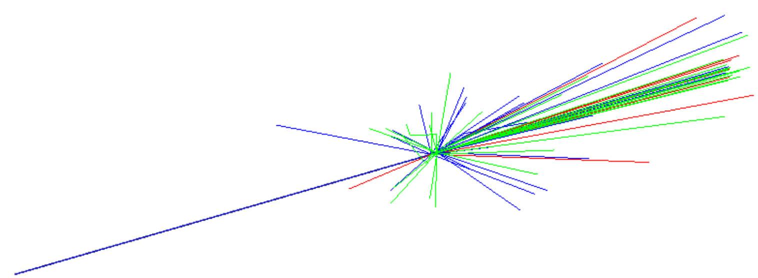


Variance Reduction - Biasing

- Even with an efficient choice of beam distribution events of interest may be rare
  - Perhaps rare due to cross-section of process
- Perhaps common but want same error bars over large energy range ⇒ variance reduction
- Classic example beam gas (interaction per event)
- Bias inelastic proton cross-section



P=1×10^{-7} bar, Nitrogen @300K

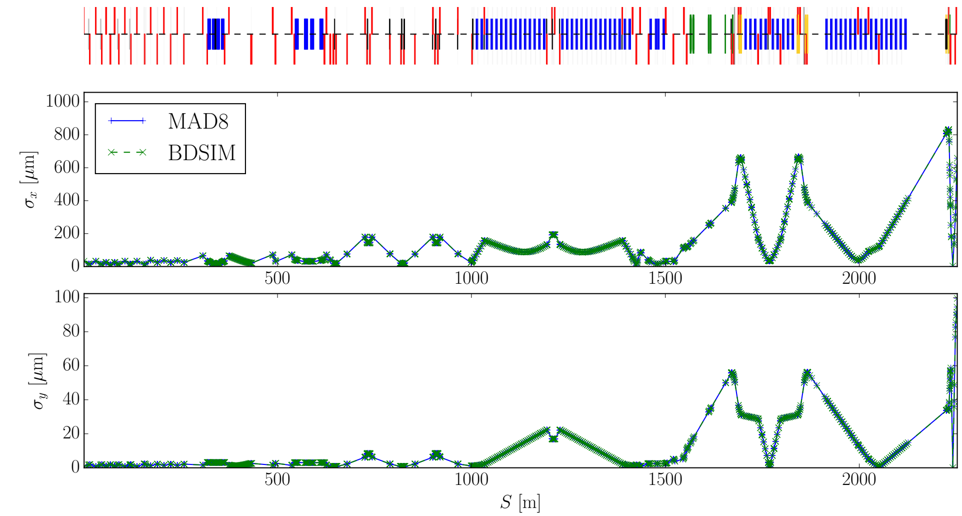


Primary interaction cross section (only) scaled by 1×10^{13}

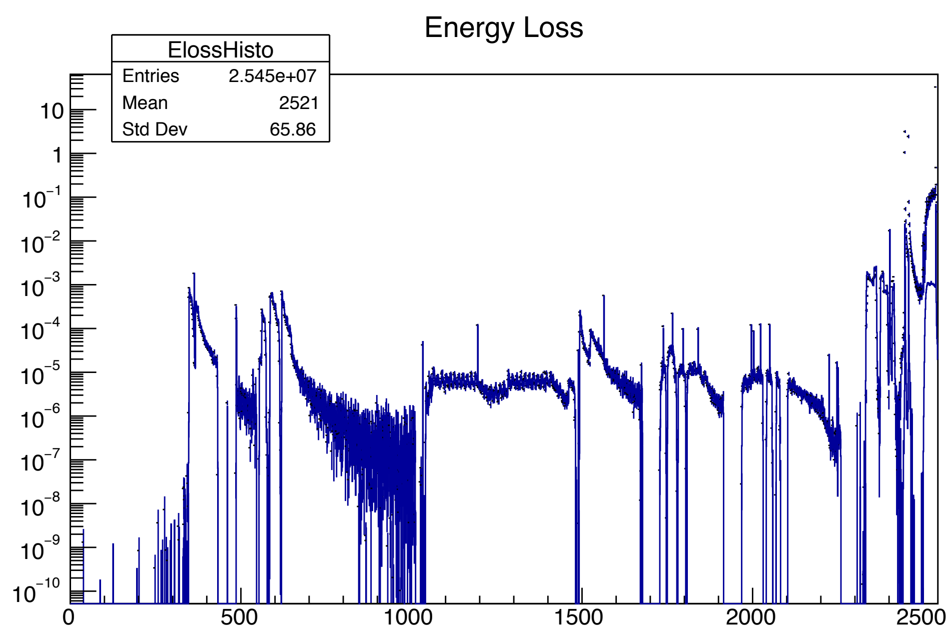
Examples

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

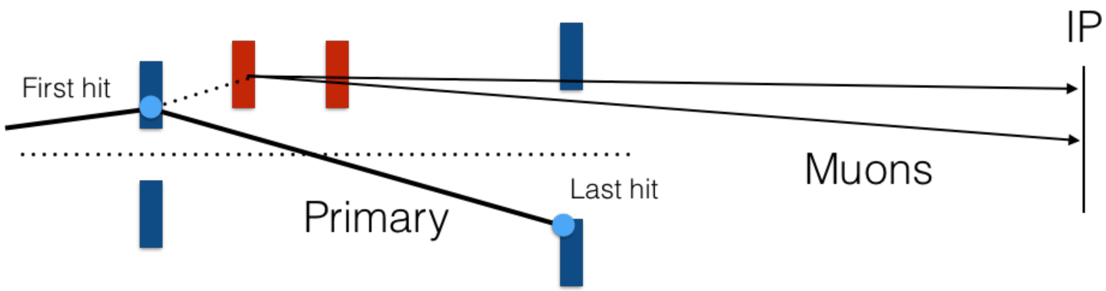


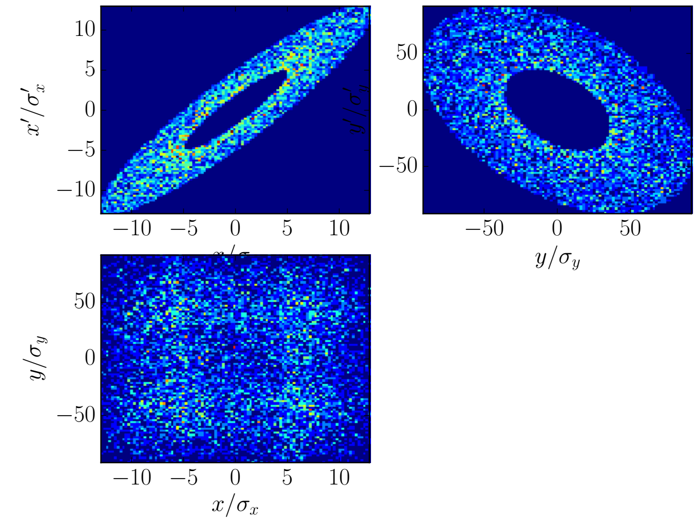
- Synchrotron radiation
  - Simple test of photon emission from all magnetic elements
  - Uses well tested and built in Geant4 SR model

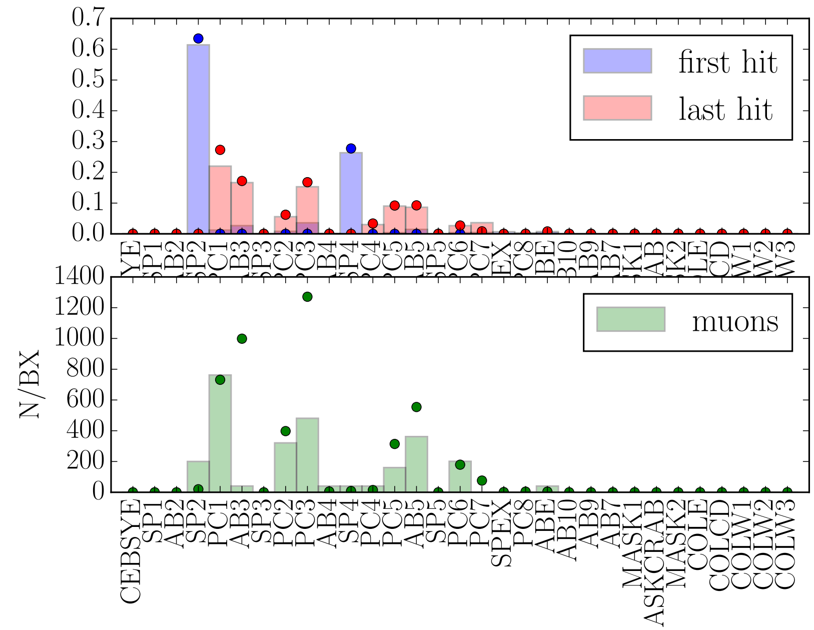


S. Boogert

ILC Muon Production Example



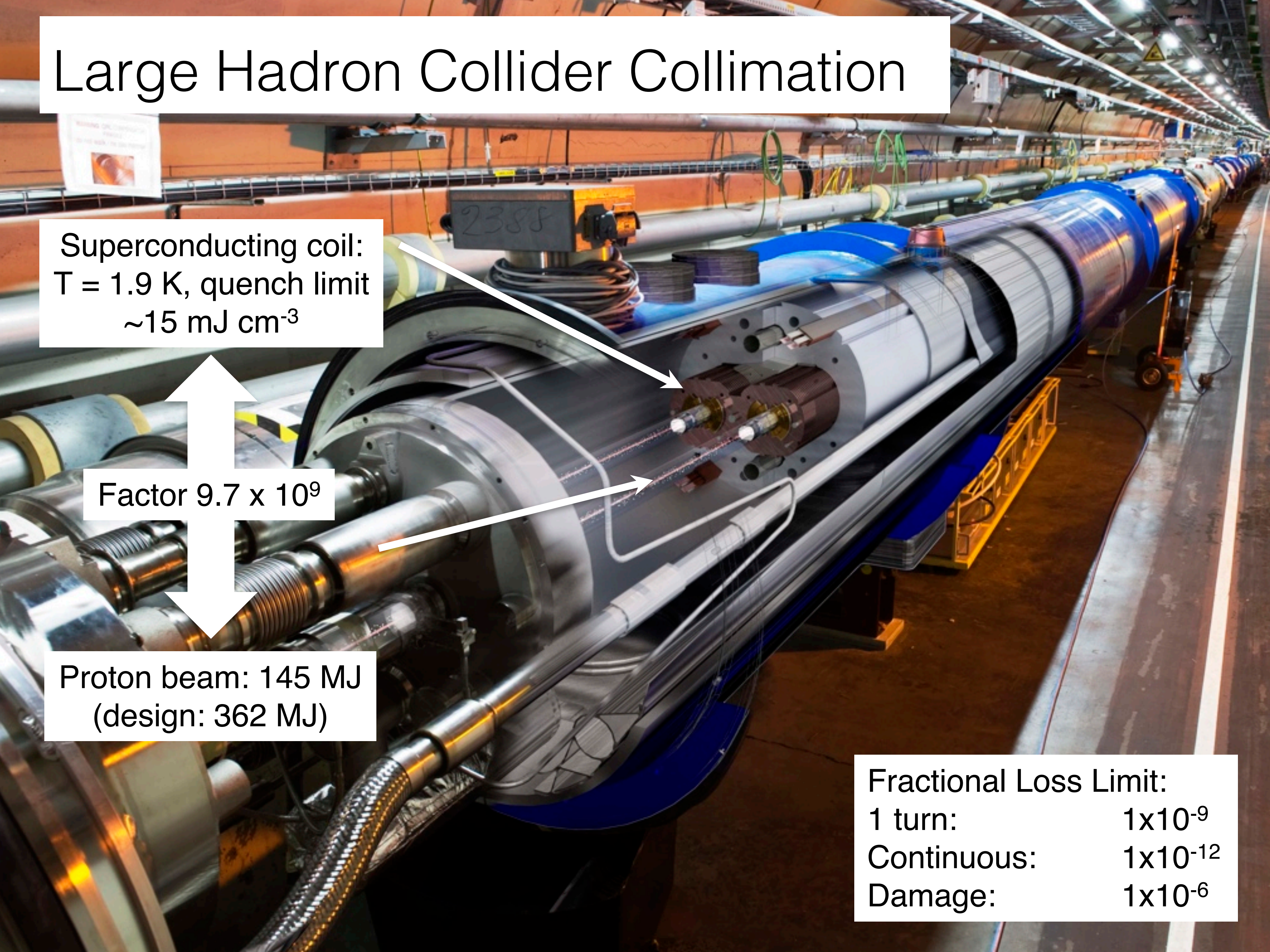




- ILC muon production interesting as large distance between production point and IP
  - Interaction in collimation system produces large number of muons
- Halo
  - 2×10^{10} electrons per bunch
  - Halo is 1×10^{-3} of total beam
  - $1/\varepsilon_{SP}$ distribution
  - x : 5 – 13 σ
  - y : 36 – 92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

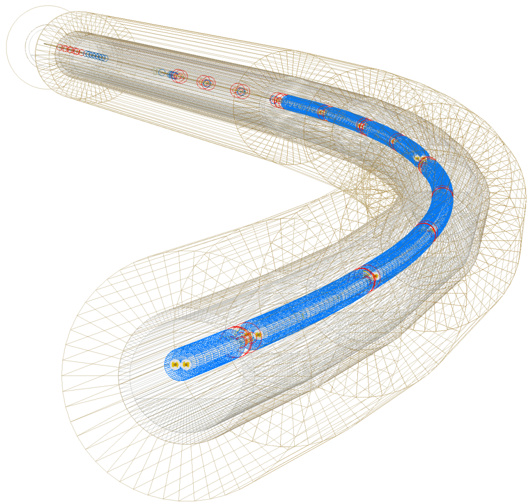
S. Boogert

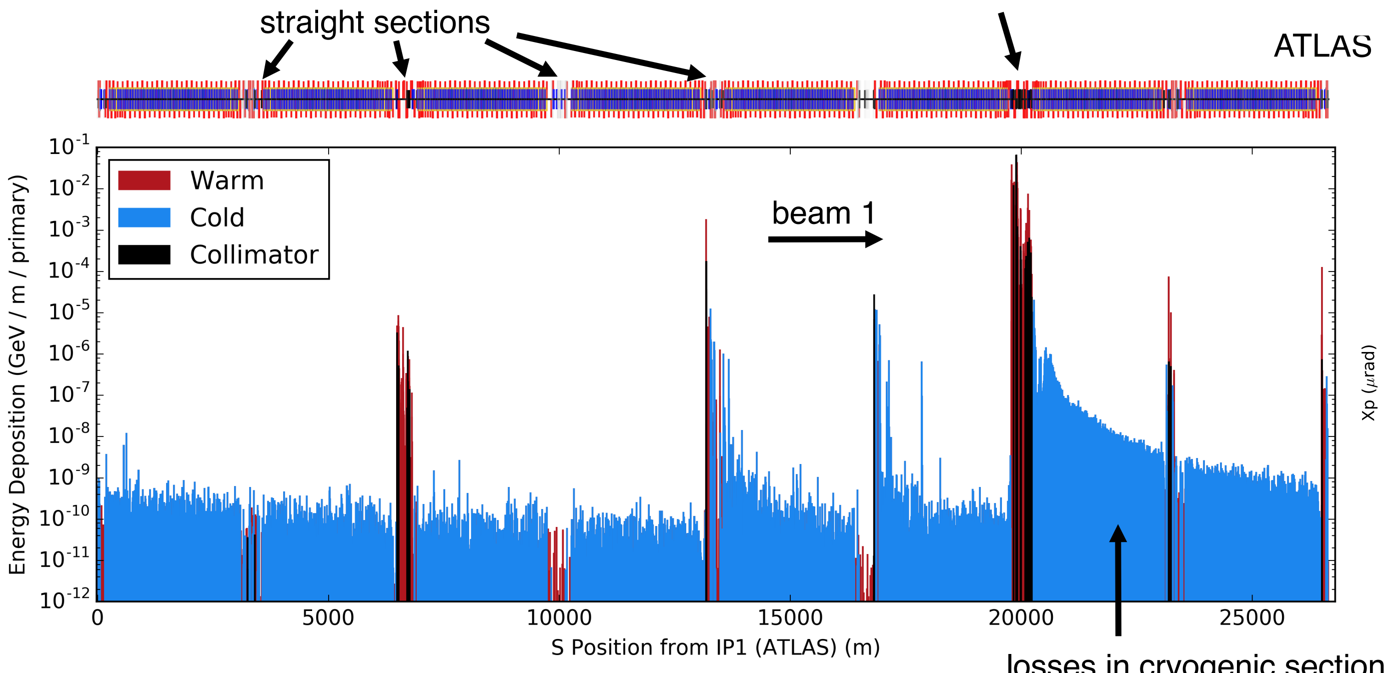
Large Hadron Collider Collimation



LHC Collimation

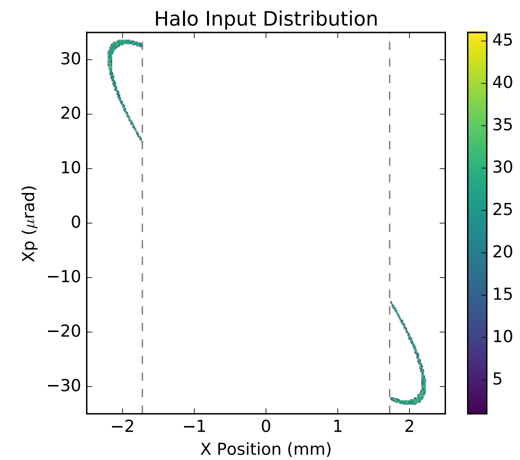
- Halo populated during beam storage
- Continually removed
- Simulate halo as it touches collimators
- LHC-style dipoles & quadrupoles
- Require 1:10^6 precision





B1 4TeV energy deposition map

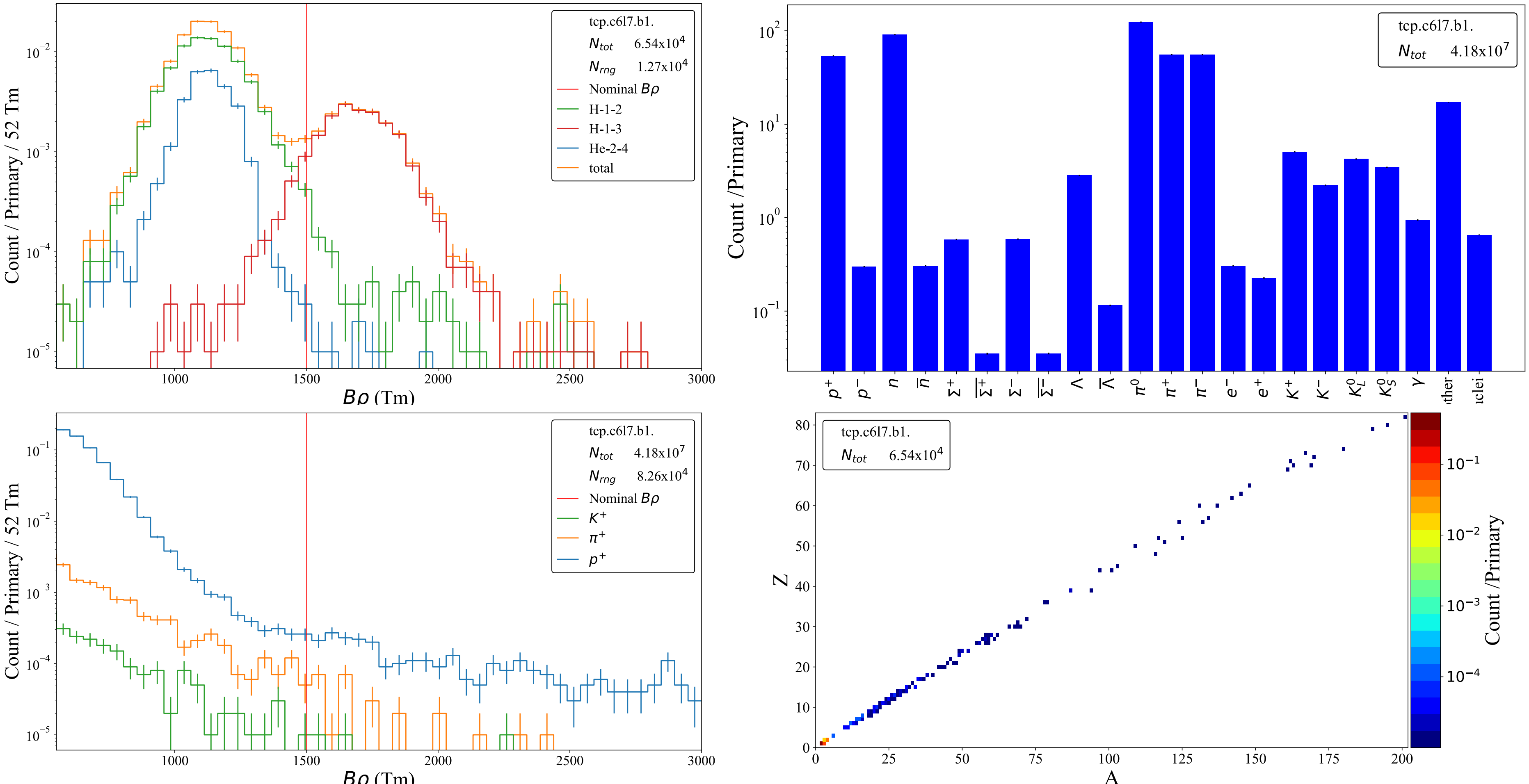
Example halo distribution



L. Nevay, S. Walker

LHC Ion Collimation

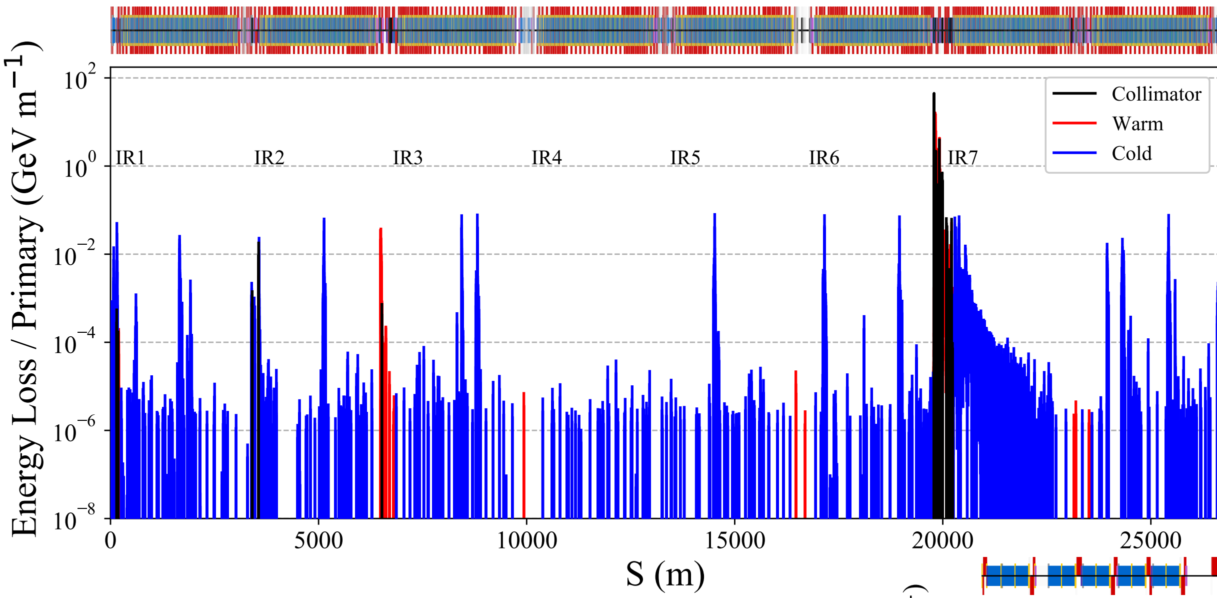
- Similarly, same model can be used with ions
- Fragmentation - many fragments around nominal Bρ



A. Abramov

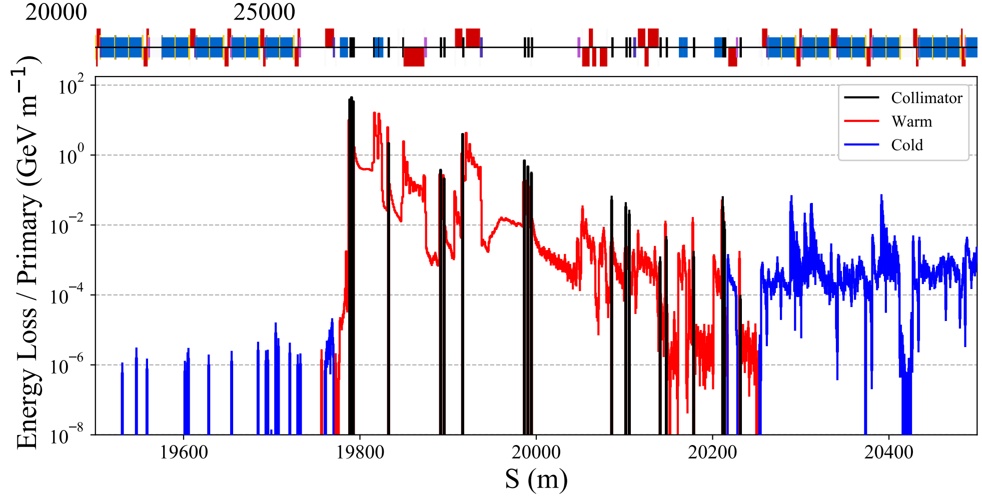
LHC Ion Collimation II

- Energy deposition around ring



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

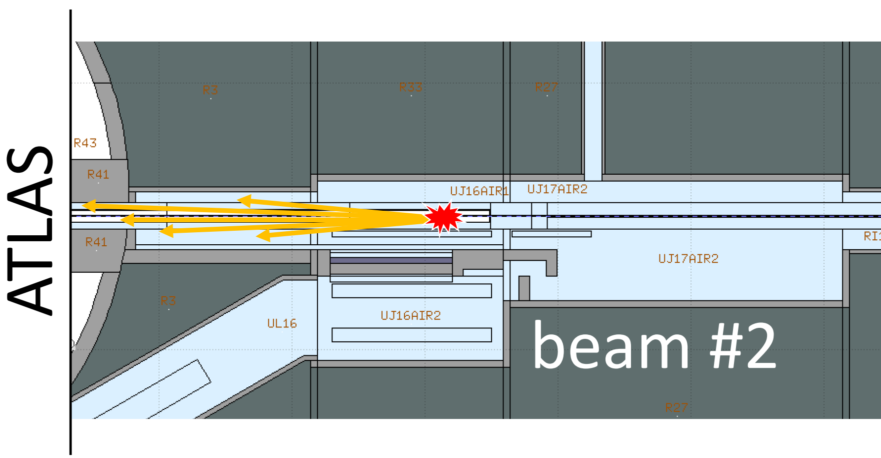
- Zoom of collimation section ('IR7')
- Coded losses on collimators, warm and cold sections

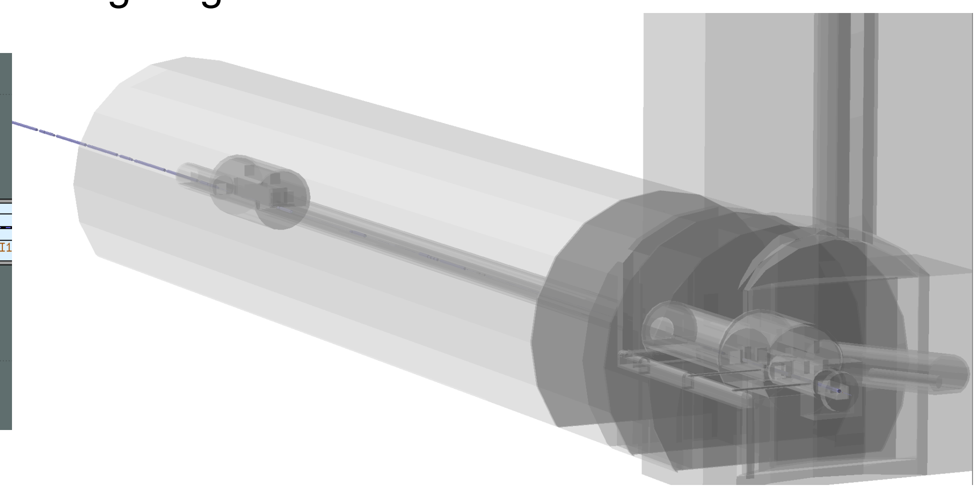


A. Abramov

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

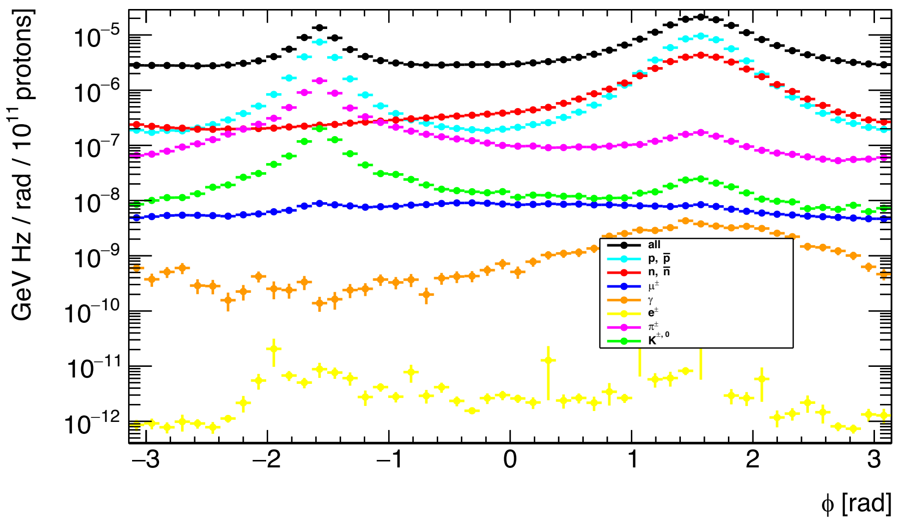




S. Walker, S. Gibson

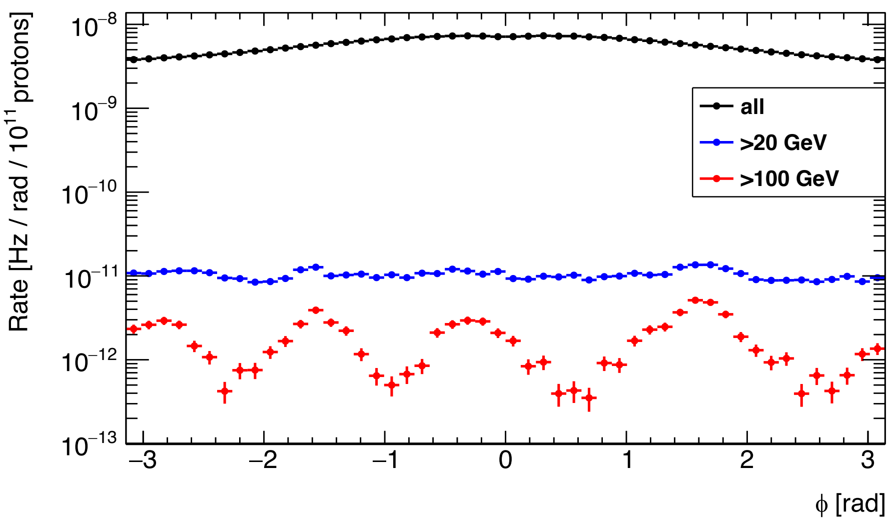
LHC Non-Collision Backgrounds II

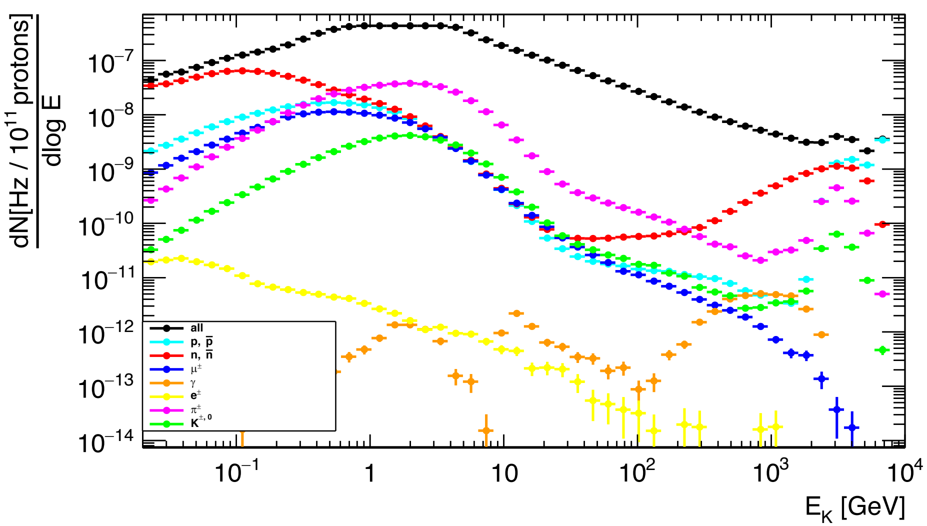
Azimuthal rate for different species



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

Azimuthal rate for different muon energies



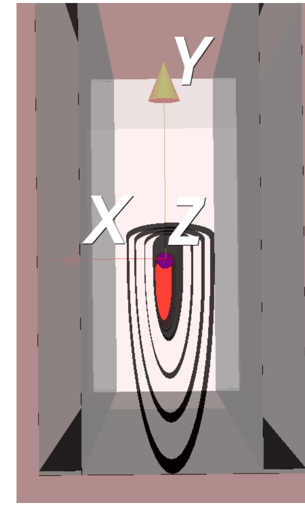


Overall particle spectra at interface plane

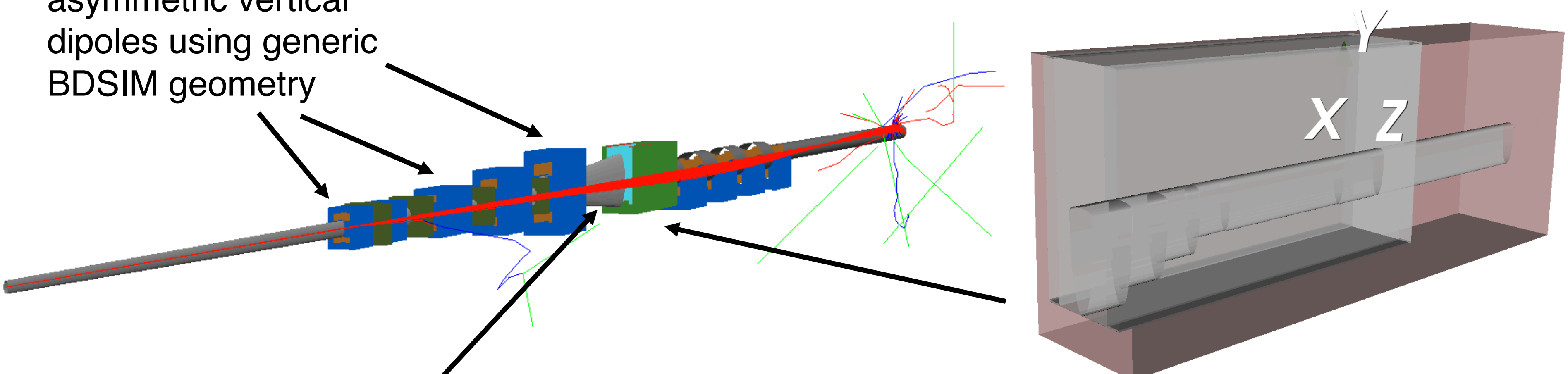
S. Walker, S. Gibson

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



asymmetric vertical dipoles using generic BDSIM geometry



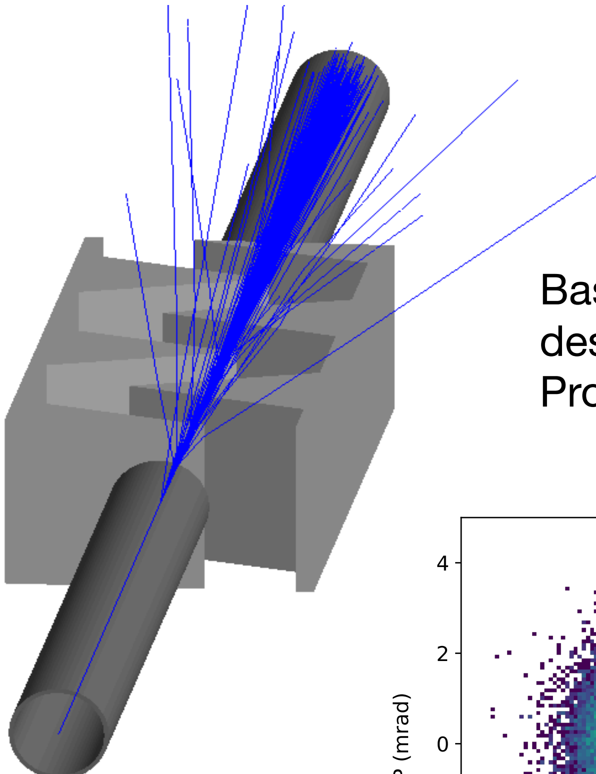
custom intermediate dump

GDML for tapered elliptical beam pipes

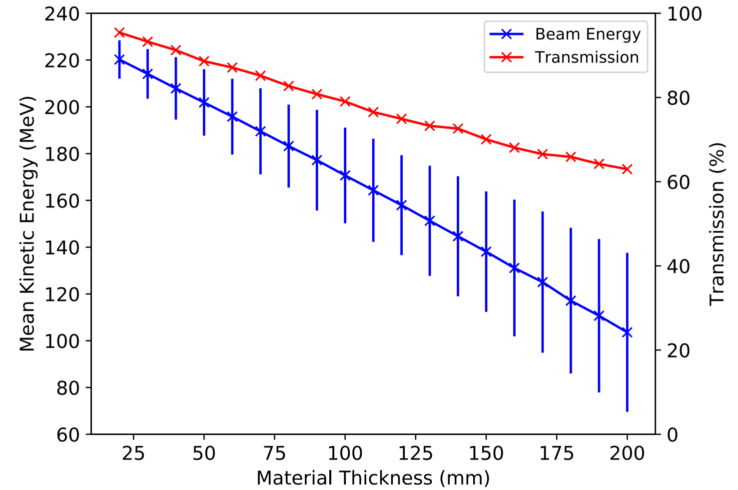
R. Bodenstein, A. Abramov

Hadron Therapy Degrader

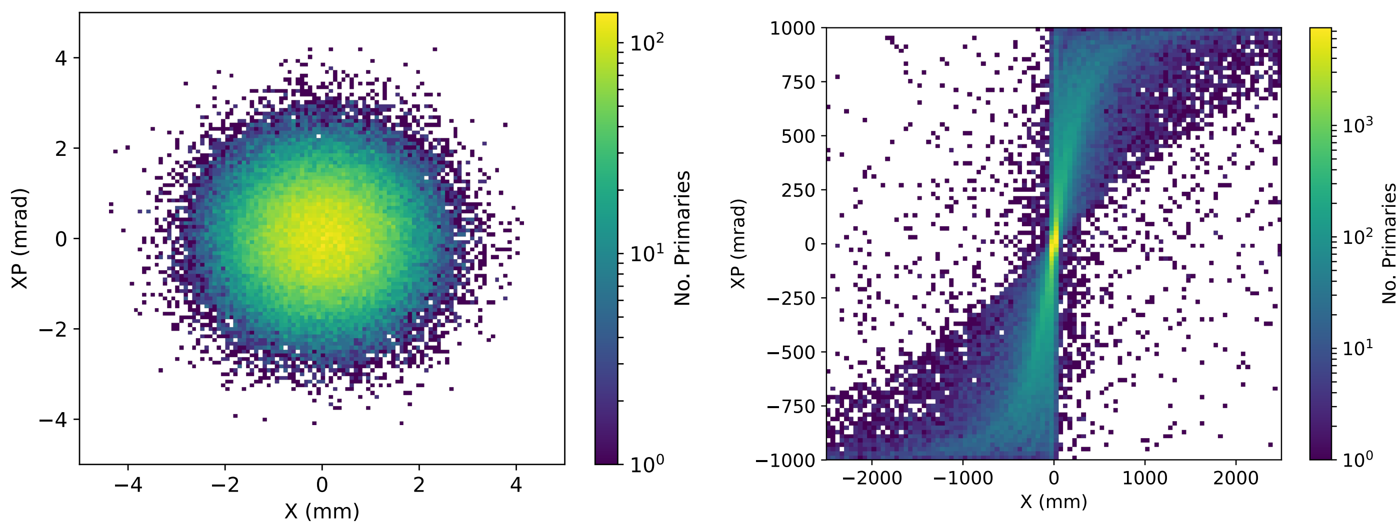
- Use variable material depth to degrade beam energy



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



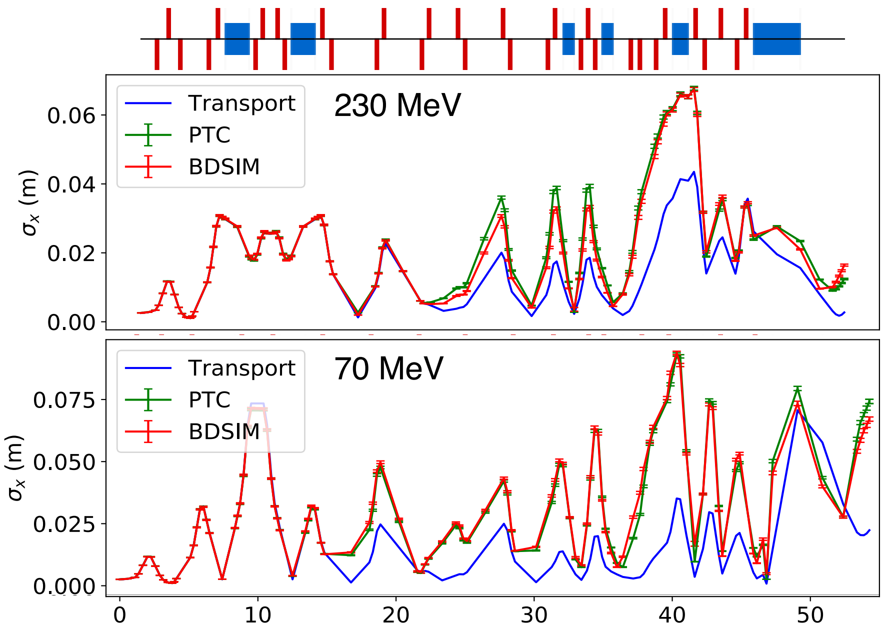
Horizontal phase space before (left) and after (right) a degrader.

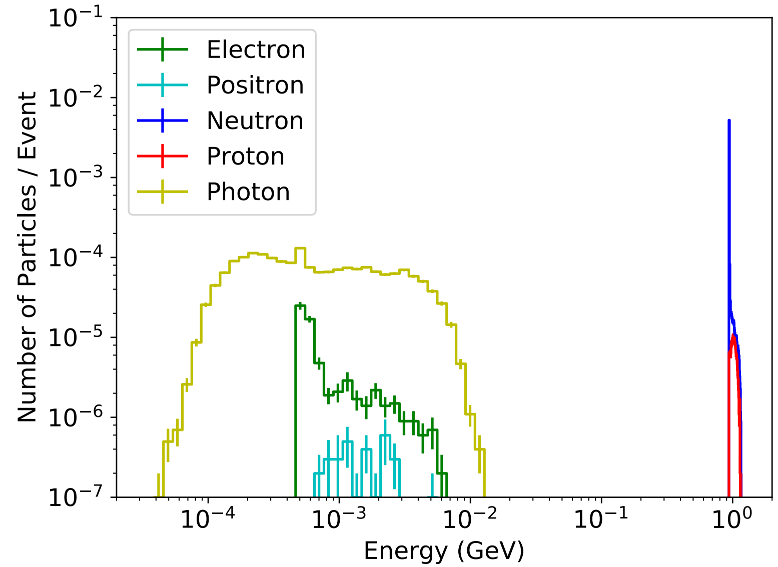


W. Shields

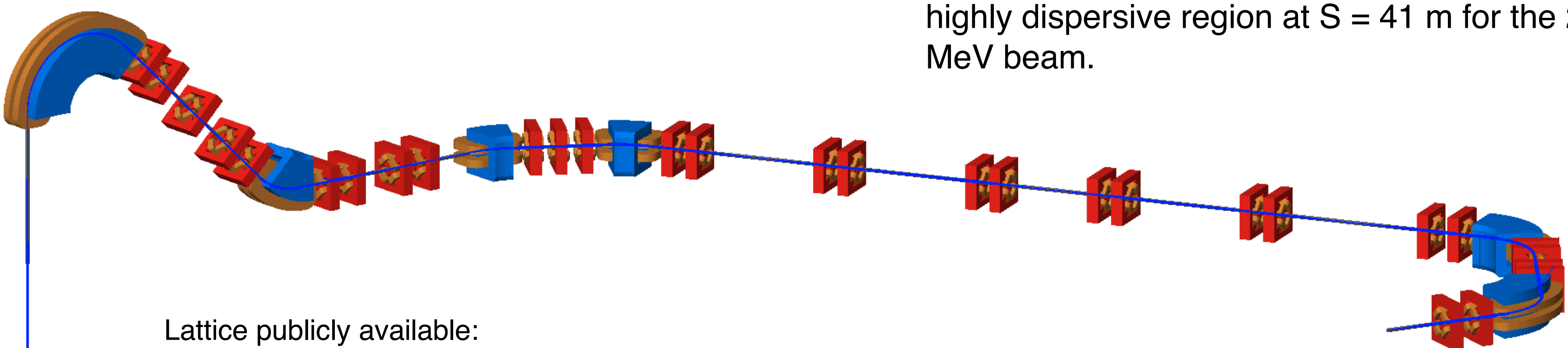
PSI Gantry II

- Optical comparison and validation





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



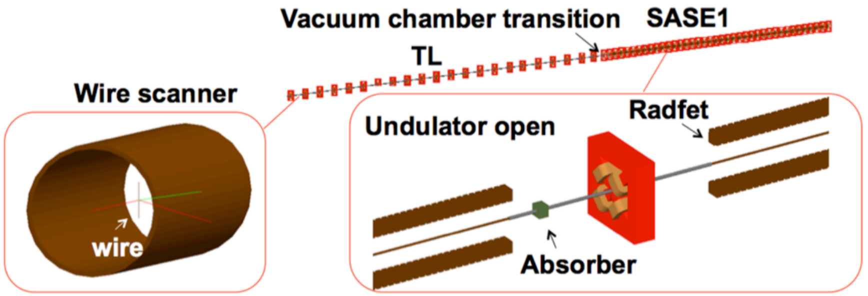
Lattice publicly available:
http://aea.web.psi.ch/Urs_Rohrer/MyFtp

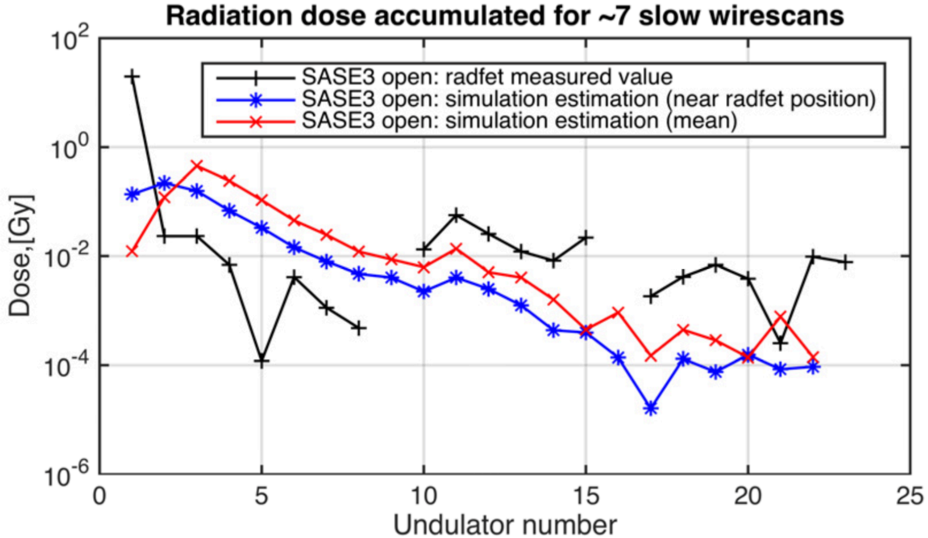
Published at IPAC 2018 - MOPML061

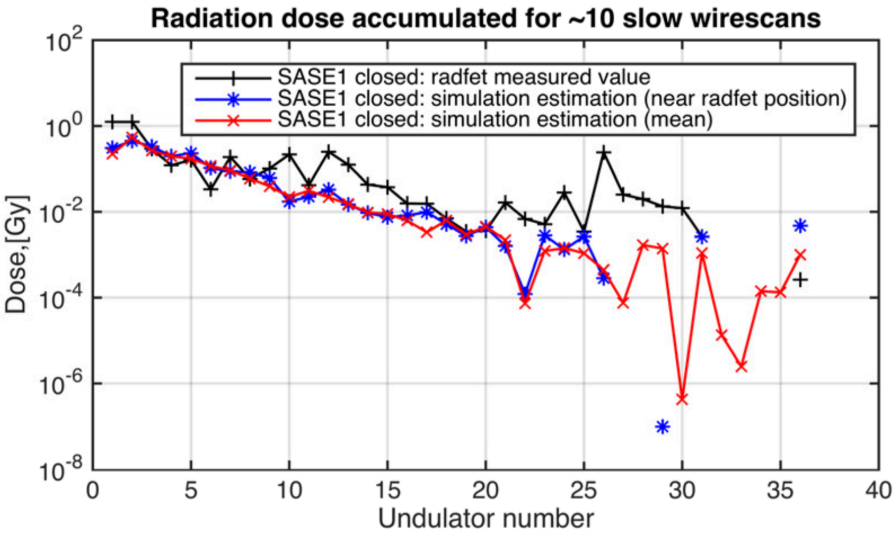
W. Shields

DESY XFEL Undulator Dose

- Undulator dose higher than original design
- Caused by secondary neutrons and synchrotron radiation
- BDSIM used to simulate dose in GDML undulator model
- Simulations compare to RADFET detectors on each undulator





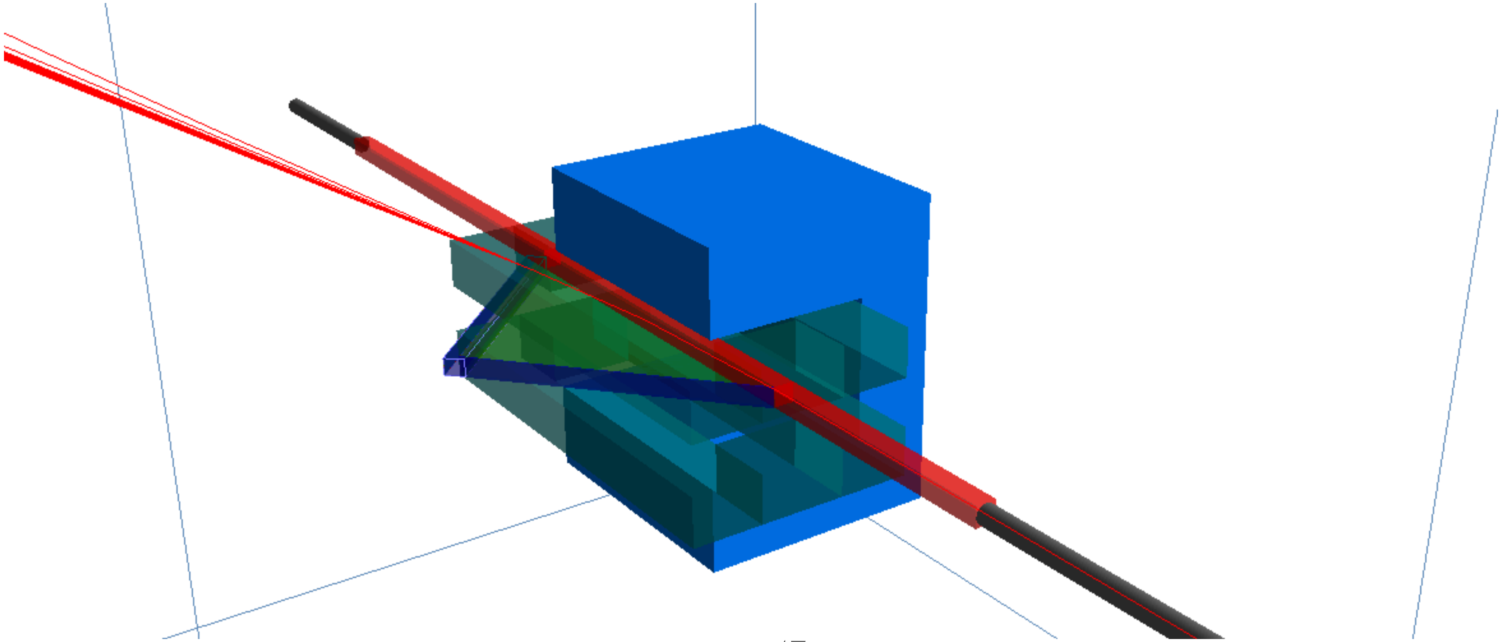


https://doi.org/10.18429/JACoW-IPAC2018-THPMF022

S. Liu, I. Agapov at DESY

AWAKE Dipole Spectrometer

- Previous developer of BDSIM 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



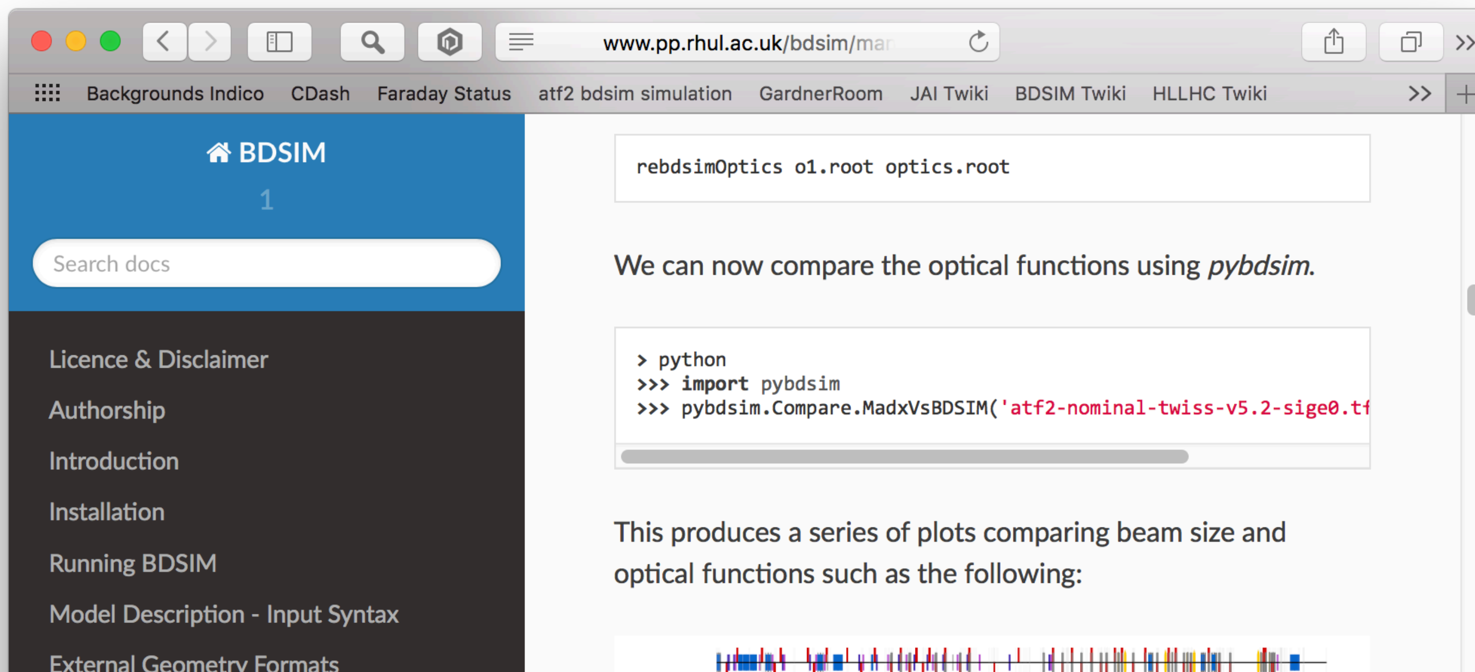
Summary

- Strategy of combined simulation demonstrated
- Spectrum from accelerator tracking to particle physics
- Radiation simulation geometry often different from realistic geometry
- BDSIM is open source C++ program containing many of these ideas
- Ready for a lot of studies, but collaboration very welcome!

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



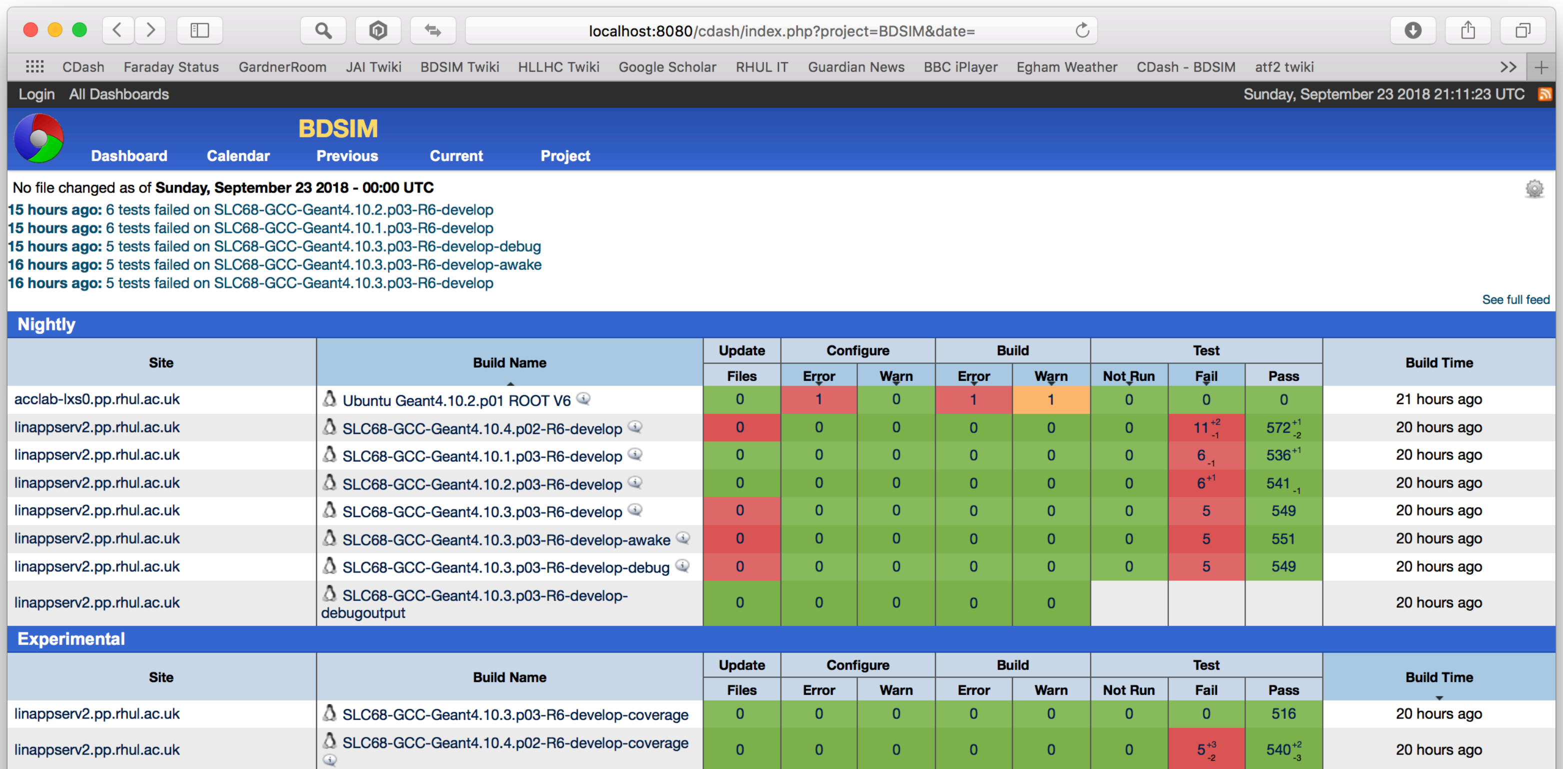
Collaborative Tools

- Public git repository
- 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

Quality & Testing

ROYAL HOLLOWAY UNIVERSITY OF LONDON

- 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



localhost:8080/cdash/index.php?project=BDSIM&date=

CDash Faraday Status GardnerRoom JAI Twiki BDSIM Twiki HLLHC Twiki Google Scholar RHUL IT Guardian News BBC iPlayer Egham Weather CDash - BDSIM atf2 twiki

Login All Dashboards Sunday, September 23 2018 21:11:23 UTC

BDSIM

Dashboard Calendar Previous Current Project

No file changed as of **Sunday, September 23 2018 - 00:00 UTC**

15 hours ago: 6 tests failed on SLC68-GCC-Geant4.10.2.p03-R6-develop
15 hours ago: 6 tests failed on SLC68-GCC-Geant4.10.1.p03-R6-develop
15 hours ago: 5 tests failed on SLC68-GCC-Geant4.10.3.p03-R6-develop-debug
16 hours ago: 5 tests failed on SLC68-GCC-Geant4.10.3.p03-R6-develop-awake
16 hours ago: 5 tests failed on SLC68-GCC-Geant4.10.3.p03-R6-develop

See full feed

Nightly

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
acclab-lxs0.pp.rhul.ac.uk	Ubuntu Geant4.10.2.p01 ROOT V6	0	1	0	1	1	0	0	0	21 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.4.p02-R6-develop	0	0	0	0	0	0	11^{+2}_{-1}	572^{+1}_{-2}	20 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.1.p03-R6-develop	0	0	0	0	0	0	6_{-1}	536^{+1}	20 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.2.p03-R6-develop	0	0	0	0	0	0	6^{+1}	541_{-1}	20 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.3.p03-R6-develop	0	0	0	0	0	0	5	549	20 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.3.p03-R6-develop-awake	0	0	0	0	0	0	5	551	20 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.3.p03-R6-develop-debug	0	0	0	0	0	0	5	549	20 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.3.p03-R6-develop-debugoutput	0	0	0	0	0				20 hours ago

Experimental

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.3.p03-R6-develop-coverage	0	0	0	0	0	0	0	516	20 hours ago
linappserv2.pp.rhul.ac.uk	SLC68-GCC-Geant4.10.4.p02-R6-develop-coverage	0	0	0	0	0	0	5^{+3}_{-2}	540^{+2}_{-3}	20 hours ago