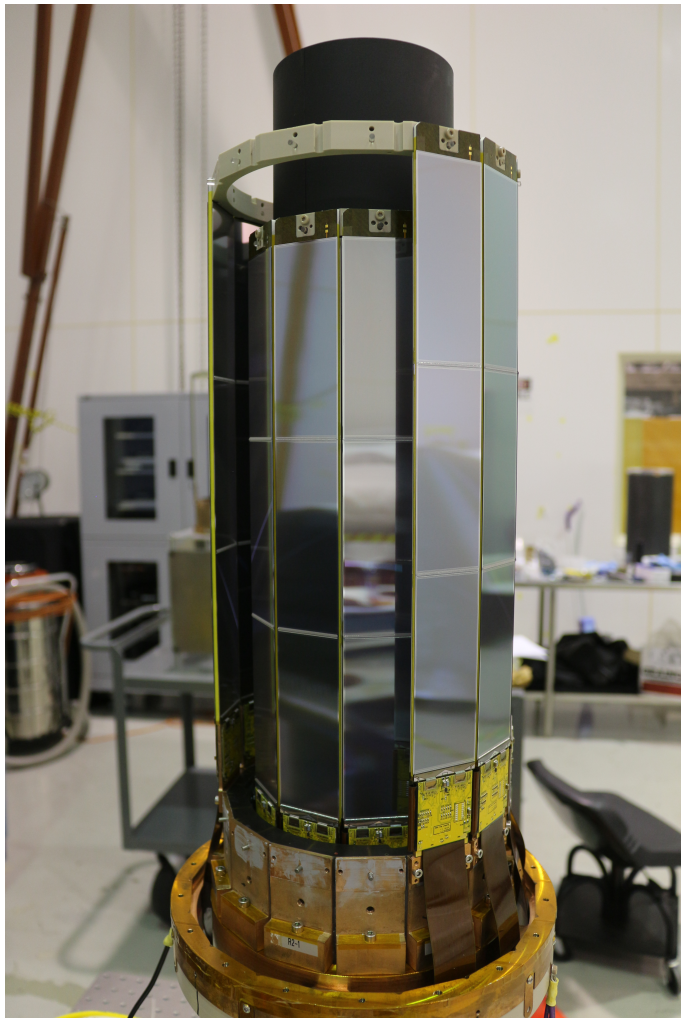
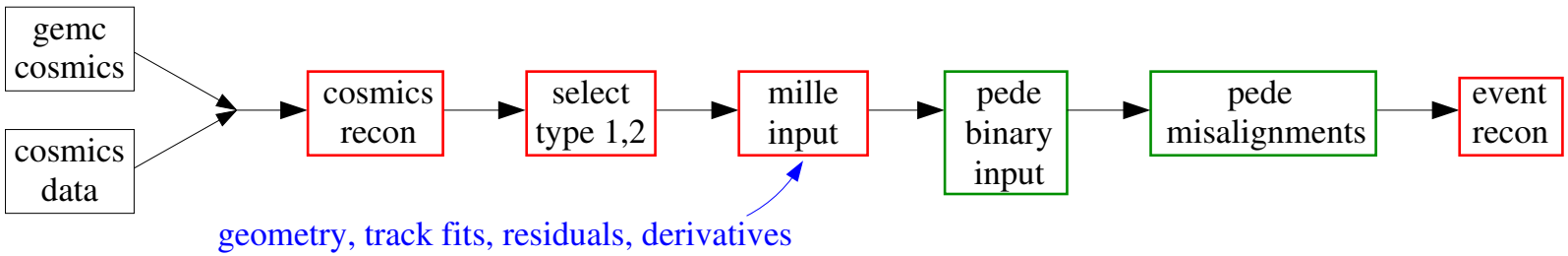
SVT Track-Based Alignment

1. Goal: Align the SVT to reach the resolution design specification of $\approx 65\ \mu m$.
2. Build accurate and complete representation of the SVT geometry and materials as part of the CLAS12 Common Tools.
3. Provide the geometry for the *gemc* simulation and the CLAS12 reconstruction from a common set of parameters.
4. Develop algorithms to measure and correct misalignments in the SVT.
5. Document it.



SVT Track-Based Alignment

1. Track-based alignment of SVT requires fitting many parameters - up to 792 here.
2. Program `millepede` does linear least squares with many parameters.
   - Matrix form of least squares method.
   - Global parameters - the geometry misalignments. Same in all events.
   - Local - individual track fit parameters. Change event-to-event.
   - Requires first partial derivatives of residuals with respect to the local (fit) parameters and global parameters (geometry misalignments).
3. Analysis chain: red boxes - Java; green boxes - C++.



4. Full chain has been tested and validated using *gemc* simulation and cosmic data for simplified case (Type 1 events).

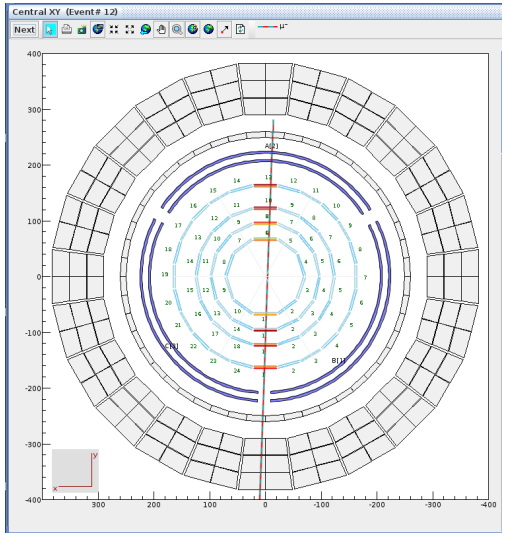
Running `millepede`

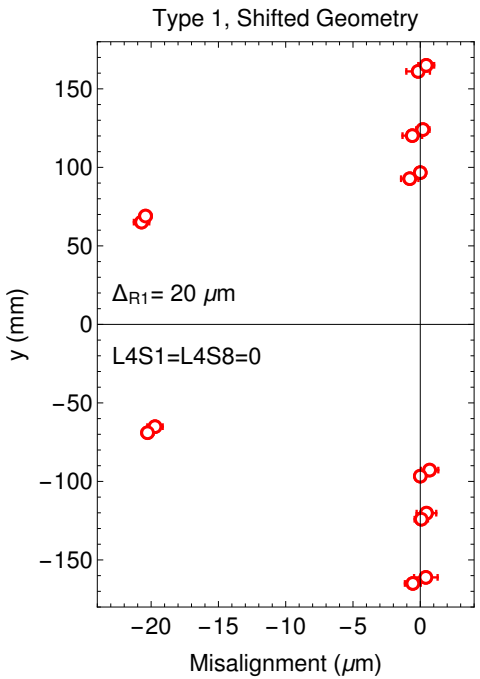
1. `mille`
   1. Code svtMille14.cc reads text file containing index, ID (layer,sector), data (residuals), derivatives (local and global), and χ^2 of fit.
   2. Use C^{++} function `mille` to generate binary input file for `pede` which does the actual fitting.
   3. Use `tools/readMilleBinary.py` to check `mille` output.
2. pede
   1. Does the actual fitting - reads binary data file from `mille`.
   2. Built with root libraries.
   3. Requires steering and constraint files.
      `/work/halld/home/mstaib/millepede/pede mp2strSVT9.txt`
   4. output files:

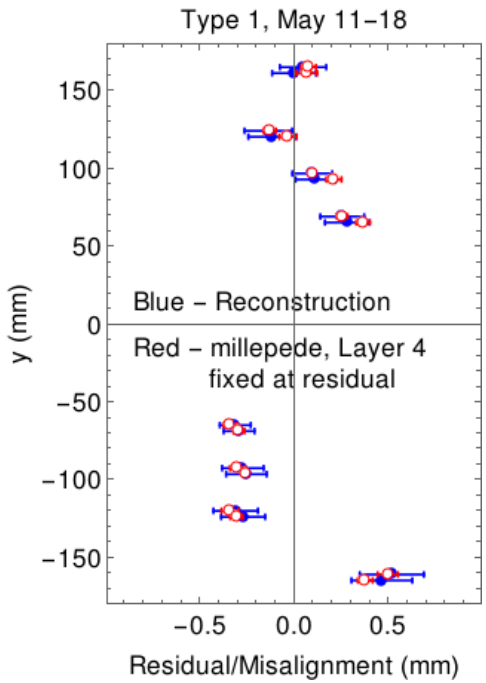
File	Purpose
millepede.log	records output
millepede.end	exit message
millepede.his	histograms
millepede.res	fit results

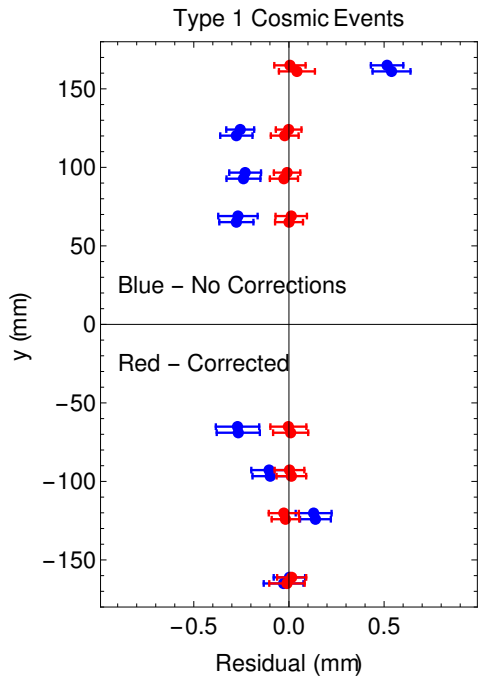
Type-1 Results

1. `millepede` applied to simulated and measured Type 1 cosmics (see ced figure).
2. Works on *gemc* cosmics with shifted regions (left-hand plot below).
3. Works on real cosmic rays collected last summer - middle and right-hand plots below.

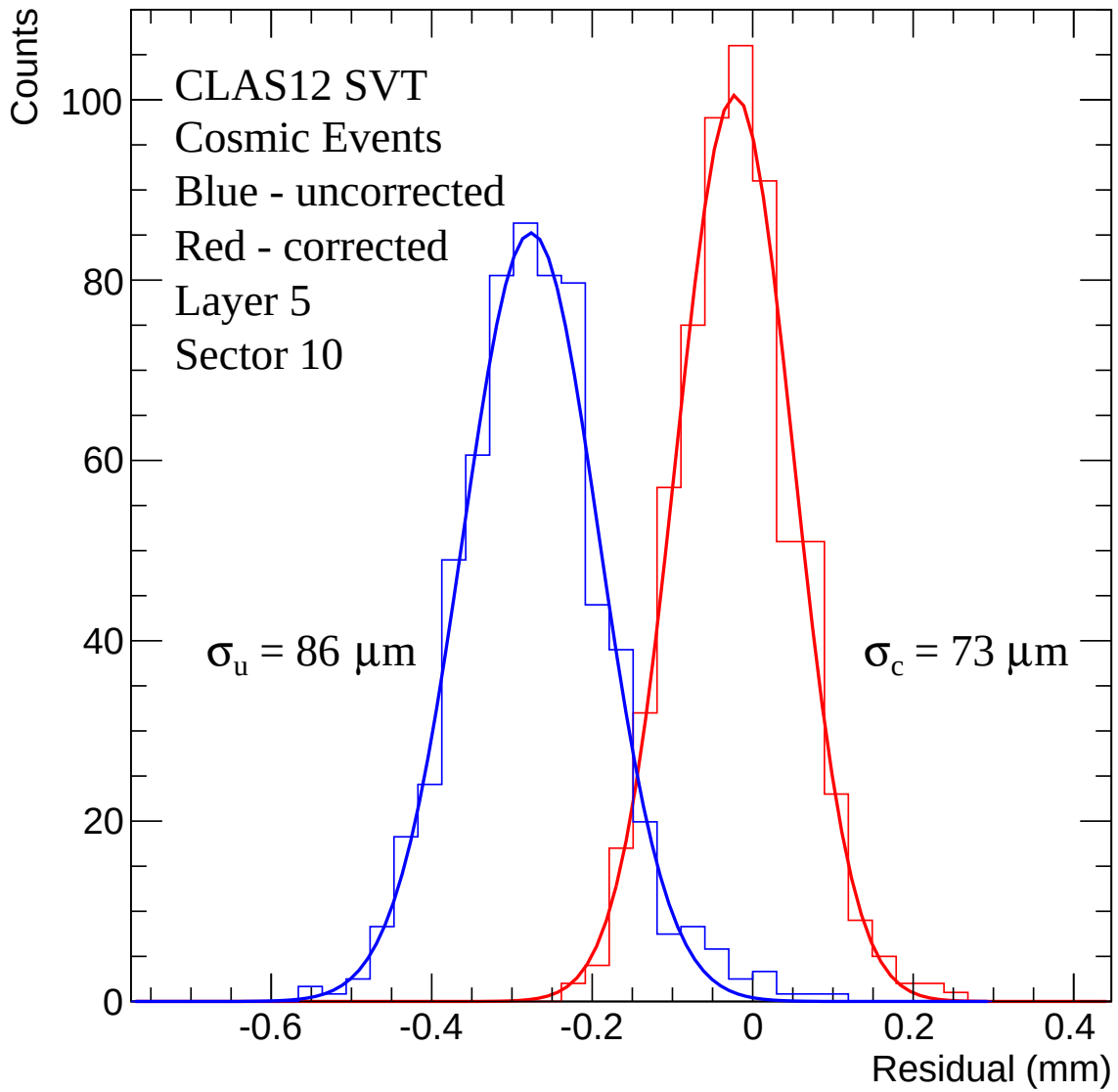






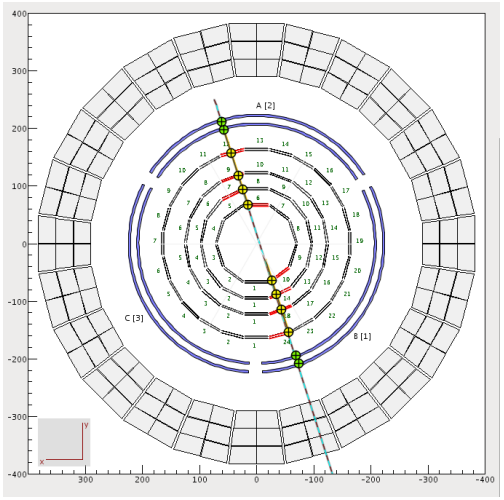


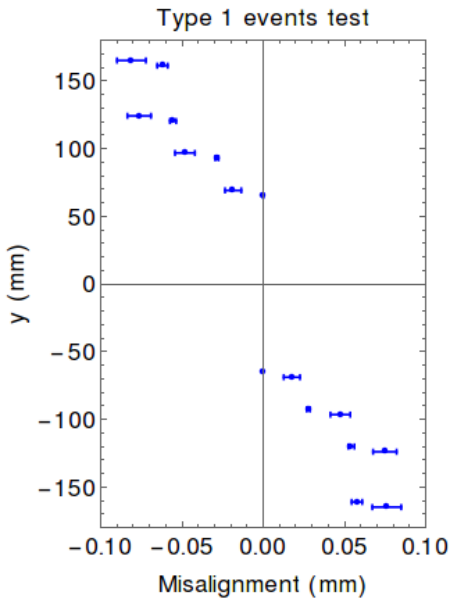
Effect on Type-1 Residuals



Status and Next Steps

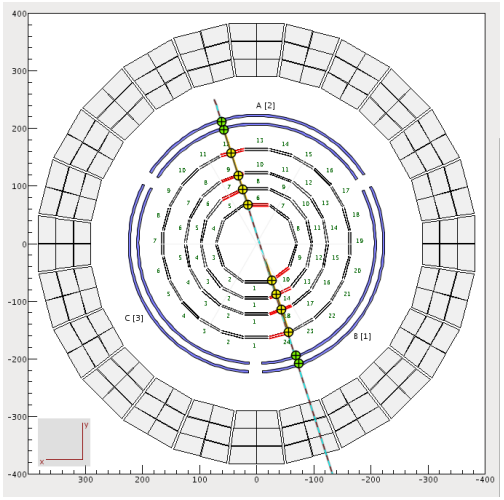
1. Status:
   - Extend to Type 2 events. Algorithm for Type 2's being tested on Type-1 events.
   - Comparison of Type-1 events analyzed with Type-2 code useful for identifying bugs, picking signs of derivatives, *etc*.
   - *gemc* version 4a.1.0 in use, Java/Groovy scripts at coatjava 4a.5.5.
2. Next steps:
   - More testing with Type-1 events.
   - Apply Type-2 to Type-2 events.
   - `millepede` codes are built for Centos 6 - need to upgrade to Centos 7.
   - Test with cosmics (simulated and measured).

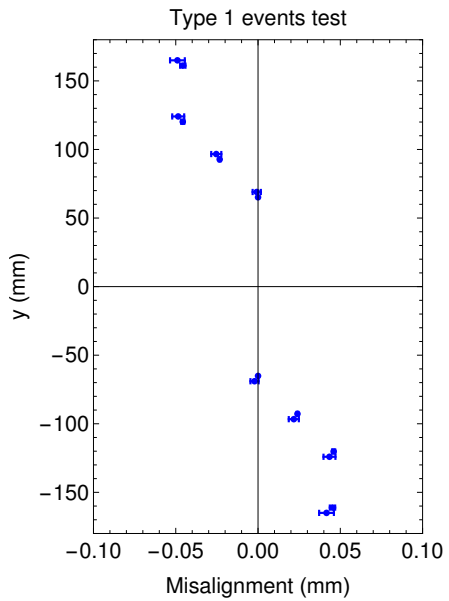




Status and Next Steps

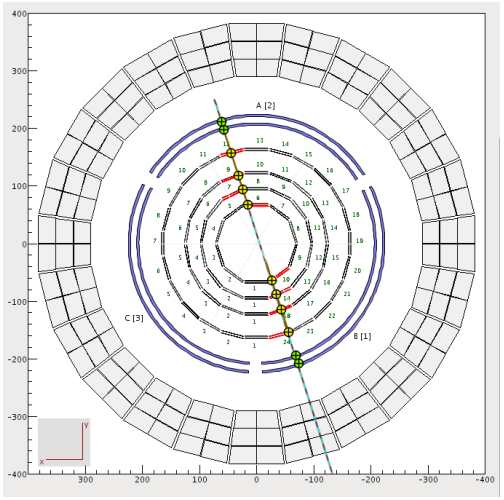
1. Status:
   - Type 2 events selected. Algorithm for Type 2's tested on Type-1 events.
   - Comparison of Type-1 events analyzed with Type-2 code useful for identifying bugs, picking signs of derivatives, *etc*.
   - *gemc* version 4a.1.0 in use, Java/Groovy scripts at coatjava 4a.5.5.
2. Next steps:
   - More testing with Type-1 events.
   - Apply Type-2 to Type-2 events.
   - `millepede` codes are built for Centos 6 - need to upgrade to Centos 7.
   - Test with cosmics (simulated and measured).

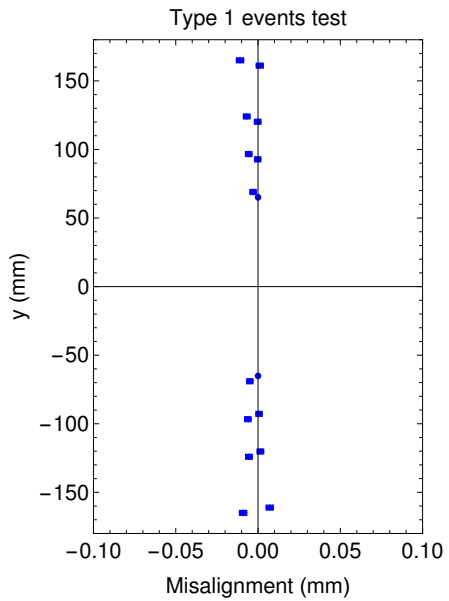




Status and Next Steps

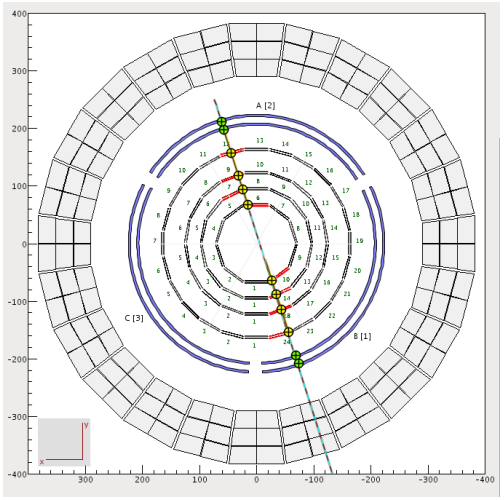
1. Status:
   - Type 2 events selected. Algorithm for Type 2's tested on Type-1 events.
   - Comparison of Type-1 events analyzed with Type-2 code useful for identifying bugs, picking signs of derivatives, *etc*.
   - *gemc* version 4a.1.0 in use, Java/Groovy scripts at coatjava 4a.5.5.
2. Next steps:
   - More testing with Type-1 events.
   - Apply Type-2 to Type-2 events.
   - `millepede` codes are built for Centos 6 - need to upgrade to Centos 7.
   - Test with cosmics (simulated and measured).

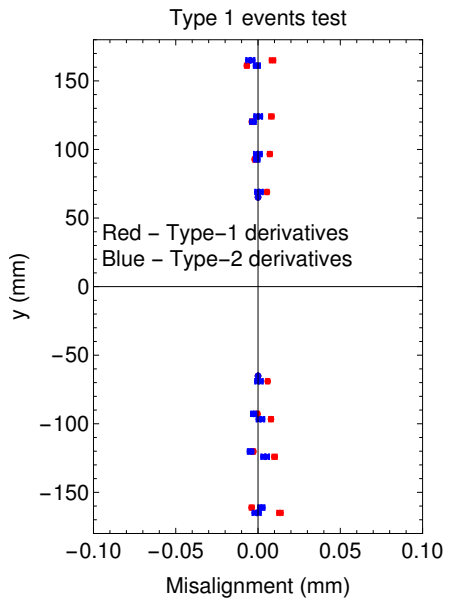




Status and Next Steps

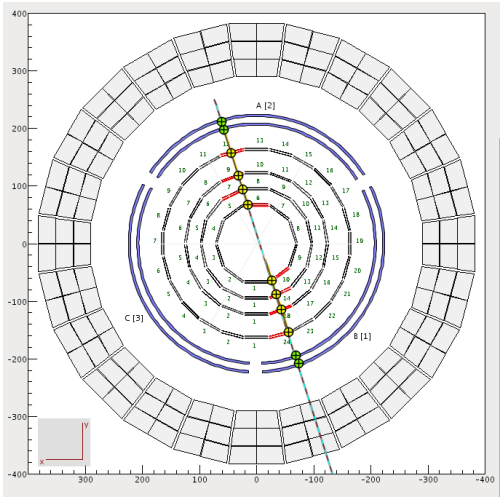
1. Status:
   - Type 2 events selected. Algorithm for Type 2's tested on Type-1 events.
   - Comparison of Type-1 events analyzed with Type-2 code useful for identifying bugs, picking signs of derivatives, *etc*.
   - *gemc* version 4a.1.0 in use, Java/Groovy scripts at coatjava 4a.5.5.
2. Next steps:
   - More testing with Type-1 events.
   - Apply Type-2 to Type-2 events.
   - `millepede` codes are built for Centos 6 - need to upgrade to Centos 7.
   - Test with cosmics (simulated and measured).

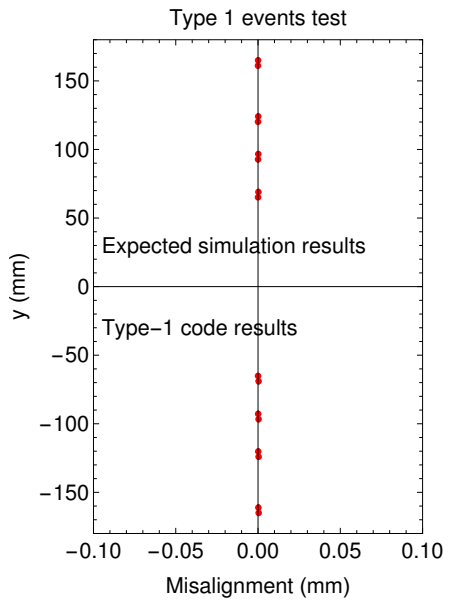




Status and Next Steps

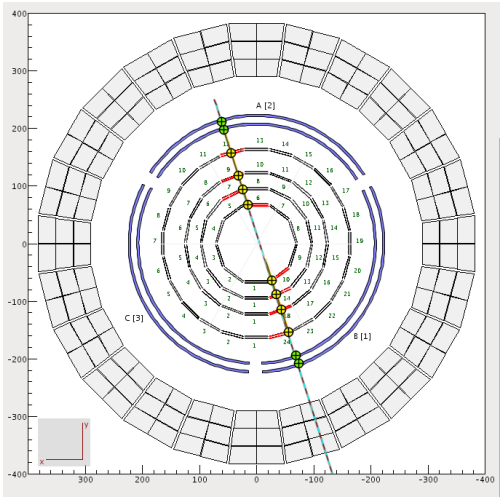
1. Status:
   - Type 2 events selected. Algorithm for Type 2's tested on Type-1 events.
   - Comparison of Type-1 events analyzed with Type-2 code useful for identifying bugs, picking signs of derivatives, *etc*.
   - *gemc* version 4a.1.0 in use, Java/Groovy scripts at coatjava 4a.5.5.
2. Next steps:
   - More testing with Type-1 events.
   - Apply Type-2 to Type-2 events.
   - `millepede` codes are built for Centos 6 - need to upgrade to Centos 7.
   - Test with cosmics (simulated and measured).

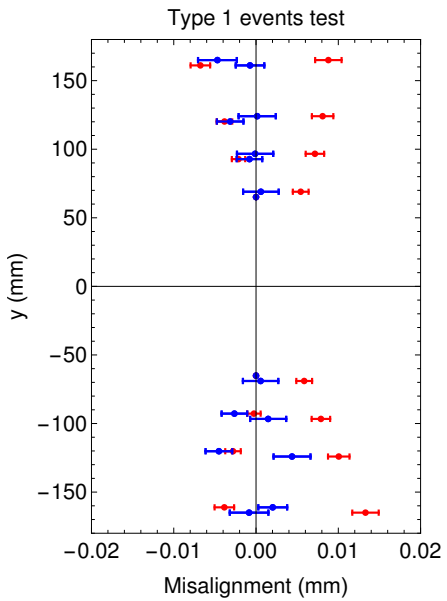




Status and Next Steps

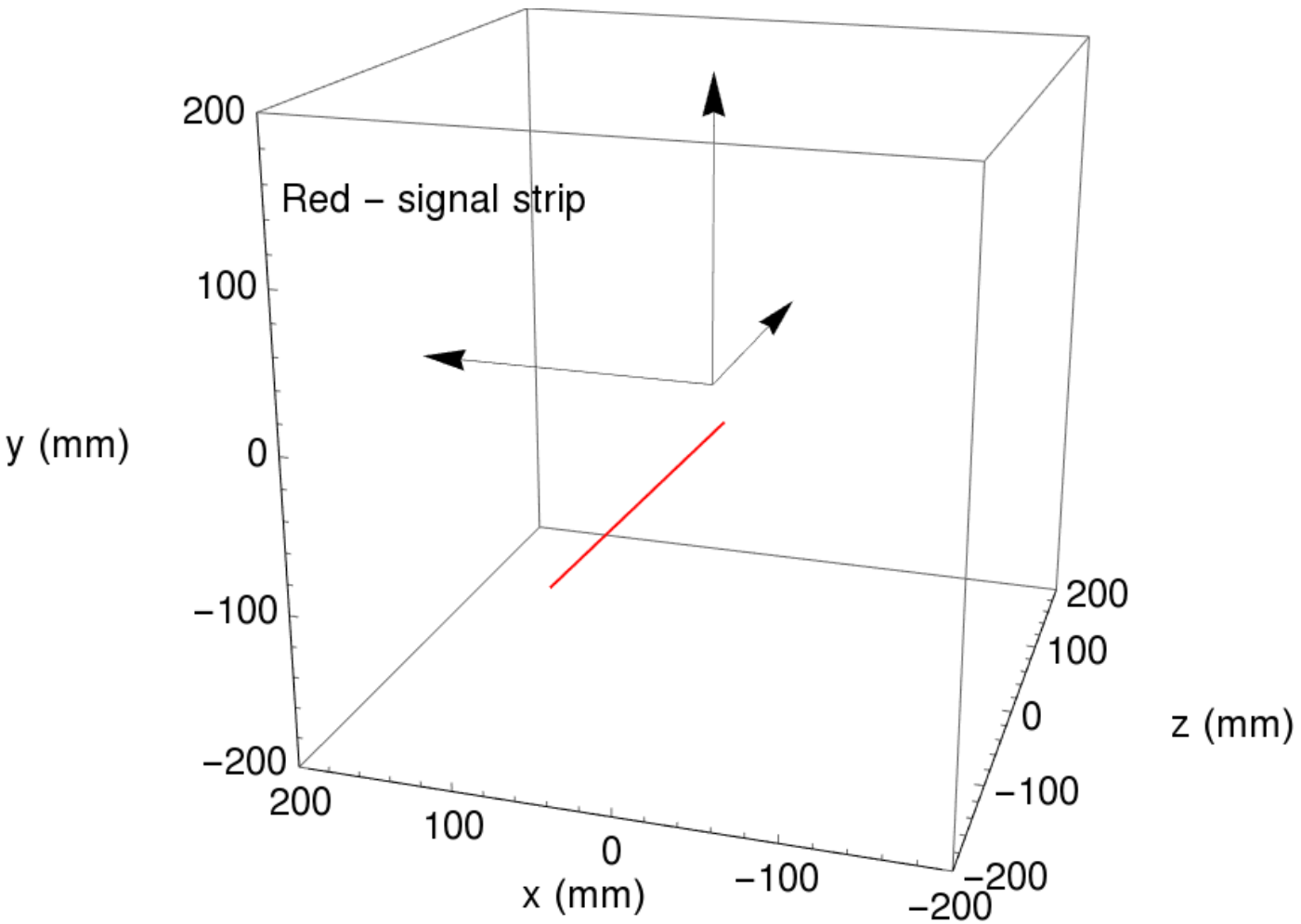
1. Status:
   - Type 2 events selected. Algorithm for Type 2's tested on Type-1 events.
   - Comparison of Type-1 events analyzed with Type-2 code useful for identifying bugs, picking signs of derivatives, *etc.*
   - *gemc* version 4a.1.0 in use, Java/Groovy scripts at coatjava 4a.5.5.
2. Next steps:
   - More testing with Type-1 events.
   - Apply Type-2 to Type-2 events.
   - `millepede` codes are built for Centos 6 - need to upgrade to Centos 7.
   - Test with cosmics (simulated and measured).
   - Extract misalignments and correct reconstruction.



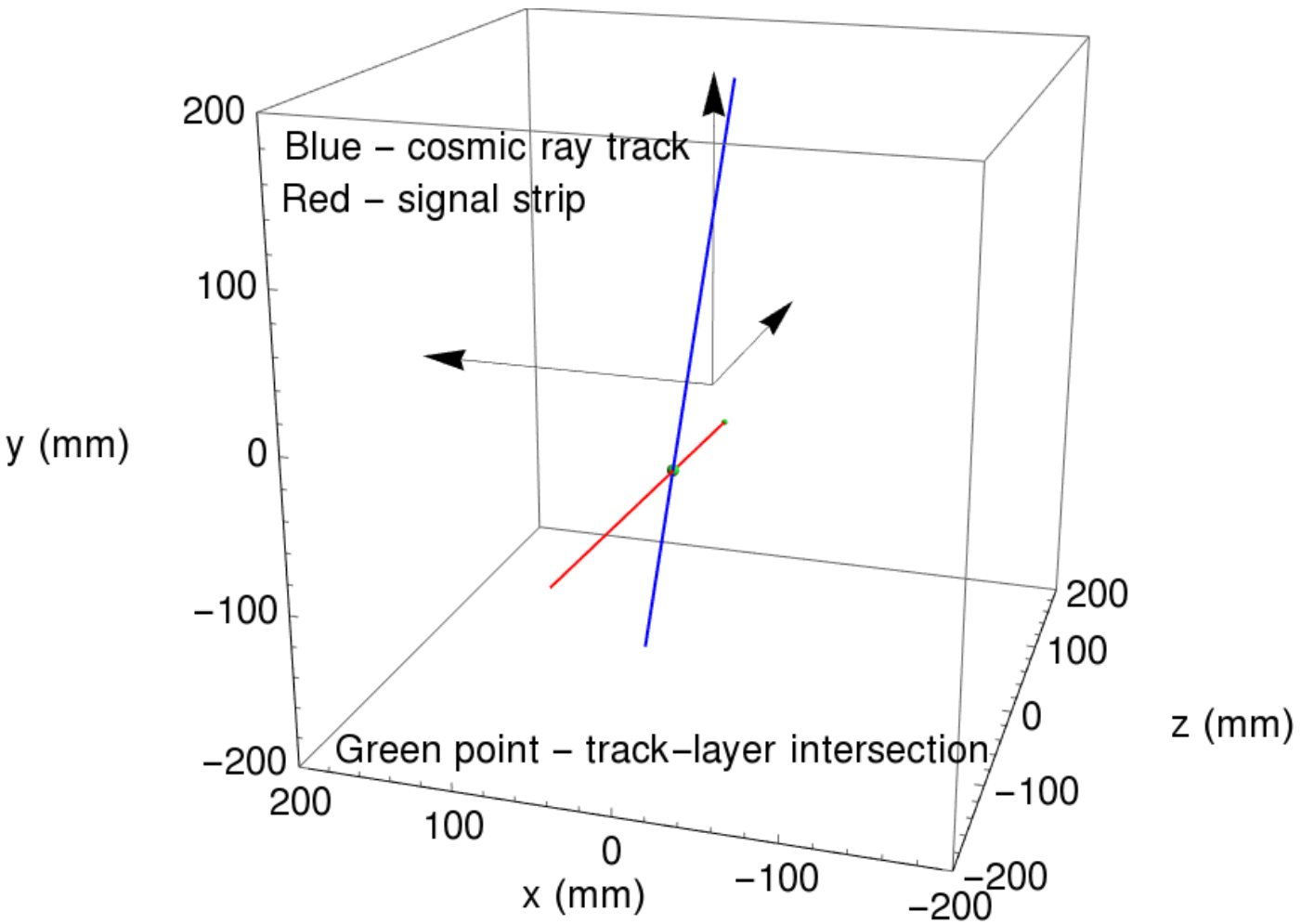


Additional Slides

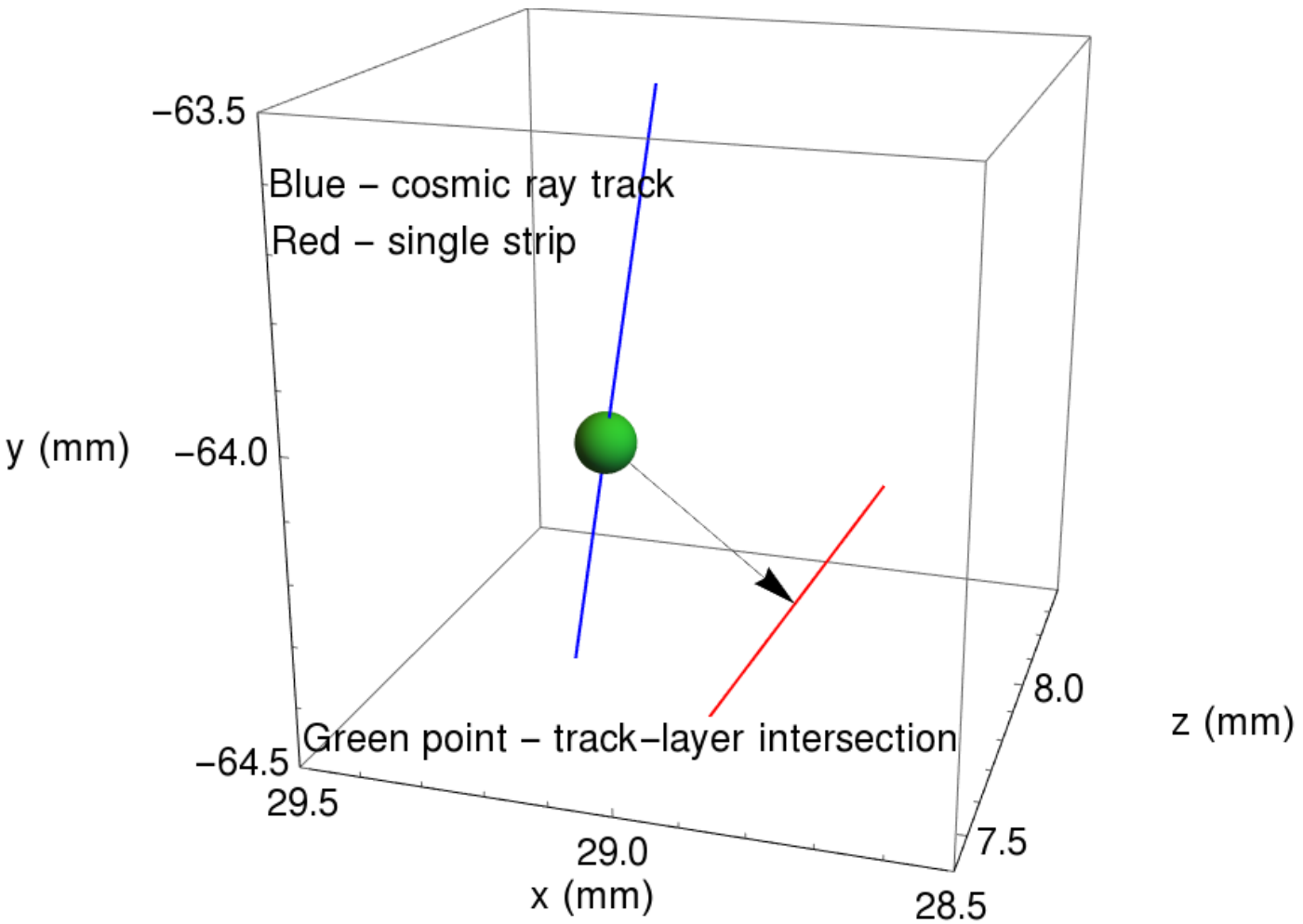
Visualization/Validation



Visualization



Visualization



Fun with *Mathematica*

$$\frac{dDOCAQ}{dmyz}$$

DOCA - distance of closest approach, *myz* - slope in $y - z$ plane.

```
[illegible]
```